MICROHOBBY SEMANAL

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR

AÑO III - N.º 67

135 PTS.
Canarias 140 ptas.

UTILIDADES

¡UNA CALCULADORA QUE FUNCIONA DENTRO DE OTROS PROGRAMAS!

IMPRESIONANTE MAPA FOTOGRÁFICO DEL "CAMELOT WARRIORS"

PROFESOR PARTICULAR

LA MEJOR AYUDA PARA RESOLVER INTEGRALES

EXPANSION

BETA BASIC 3.Ø: El lenguaje más potente para Spectrum



HOBBY PRESS

Director Editorial
José I. Gómez-Centurión
Director Ejecutivo
Domingo Gómez
Asesor Editorial
Gabriel Nieto
Redactor Jefe
Africa Pérez Tolosa
Diseño
Rosa María Capitel
Redacción
Amalio Gómez, Pedro Pérez, Jesús Alonso
Secretaría Redacción
Carmen Santamaría
Colaboradores
Primitivo de Francisco, Rafael Prades, Miguel Sepúlveda Sergio Martínez y J. M. Lazo
Corresponsal en Londres
Alan Heap
Fotografía
Javier Martínez, Carlos Candel
Portada
José María Ponce
Dibujos
J. R. Ballesteros, A. Perera, F. L. Frontán, Pejó, J. M. López Moreno, J. Igual, J. A. Calvo, Lóriga, J. Olivares

Edita
HOBBY PRESS, S. A.
Presidente
María Andrino
Consejero Delegado
José I. Gómez-Centurión
Jefe de Publicidad
Marisa Esteban
Publicidad Barcelona
José Galán Cortés
Tels.: 303 10 22 - 313 71 76
Secretaría de Dirección
Marisa Cogorro
Suscripciones
M.ª Rosa González
M.ª del Mar Calzada
Redacción, Administración y Publicidad
La Granja, 39
Polígono Industrial de Alcobendas
Tel.: 654 32 11
Telex: 49480 HOPR
Dto. Circulación
Carlos Peropadre
Distribución
Coedis, S. A. Valencia, 245
Barcelona
Imprime
Rotedic, S. A. Ctra. de Irún, km. 12,450 (MADRID)
Fotocomposición
Espacio y Punto, S. A.
Paseo de la Castellana, 268
Fotomecánica
Grof
Ezequiel Solana, 16
Depósito Legal
M-36.598-1984

Representante para Argentina, Chile, Uruguay y Paraguay, Cia. Americana de Ediciones, S.R.L. Sud América 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina).

MICROHOBBY ESTA SEMANA

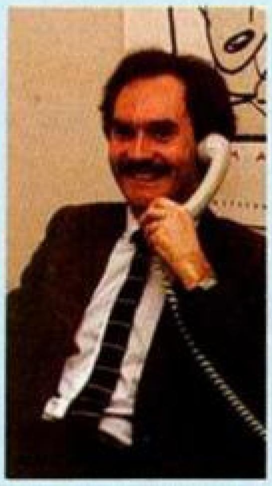
AÑO III. N.º 67. 25 de febrero al 4 de marzo de 1986
135 ptas.





Todo un despliegue de información sobre el «Camelot Warriors». Pág. 11.

AQUI LONDRES



Dos recientes encuestas en el mercado de micros han confirmado que, una vez más, *Sinclair* se ha llevado la mejor tajada en el mercado de hardware con un total, aproximado del 37% del total de las ventas durante 1985.

Sin embargo, existen diferencias entre las dos encuestas en lo que respecta al resto de la lista, puesto que ellas sitúan a Amstrad en el n.° 2, Acom n.° 3 y Commodore n.° 4, mientras que la otra cataloga a Commodore como n.°2, Acom n.° 3 y Amstrad n.° 4.

Estas discrepancias se pueden explicar por el gran número de máquinas baratas que Commodore y Acom lanzaron al mercado poco antes de la avalancha de Navidades, y el gran número de ventas de estos micros ha deformado, en cierto modo, las cifras del mercado.

Se estima que el n.° de unidades de micros vendidos bajó un 17% en 1985, alrededor de 1,1 millones de unidades, y se predice que el n.° de ventas será de unos 750.000 en 1986, si bien el valor real del mercado aumentará en unos 262 millones de libras debido a la subida de precios de los micros más recientes como el Atari 520ST, el Amiga, el C128 y los ordenadores Amstrad.

Buenas noticias para los poseedores de ordenadores de 16 bits. *Activision,* una de las compañías más importantes de software, ha anunciado su decisión de lanzar software de 16 bits que serán versiones de los programas de C64: Hacker, Mind Shadow, y Music Studio las cuales han sido mejoradas y convertidas para el Atari 520ST y para el Amiga. Las dos primeras también serán asequibles para el Machintosch.

Otra cinta con fines benéficos está siendo creada por *War on Want,* la organización benéfica que recauda fondos para los pobres, particularmente en el Tercer Mundo.

Se espera que la cinta contenga 10 juegos y que se venda por 10£. Hasta el momento varias compañías han acordado tomar parte y donar juegos para la producción de esta cinta, entre los que se incluye Ocean, Firebird, Virjin, Atic, Quicksilva y PSS. La cinta se llamará *«Wow Ganes»* y estará disponible en principio para el Spectrum, pero si la demanda es grande se buscarán versiones para C64 y Amstrad. El lanzamiento está proyectado para el próximo mes de abril.

Timex Portujal pronto anunciará un nuevo sitema de disco para Spectrum. Lleva integrado su propio chip Z80 y dos disk-drive de 3 pulgadas (como el Amstrad). Saldrá con CTM y tendrá dos port de impresora.

De nuestro corresponsal en Londres ALAN HEAP

LENSLOCK

UN DISPOSITIVO CONTRA LA PIRATERIA

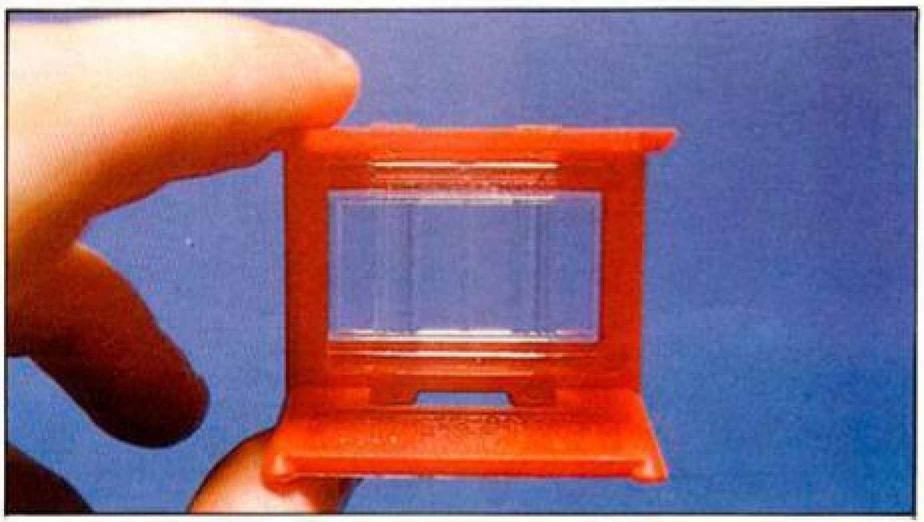
En números anteriores dimos a conocer la noticia de que ASAP había diseñado un revolucionario sistema de protección de programas llamado Lenslock, el cual consiste en una pequeña lente que, colocada a una cierta distancia de la pantalla, permite descifrar un código que, a simple vista, resulta completamente ilegible.

Al parecer, esta lente había creado algunos problemas en Gran Bretaña debido a que el manual de instrucciones que le acompañaba resultaba un tanto ambiguo y no especificaba correctamente la manera adecuada de utilizarlo.

Microhobby ha tenido acceso a una de estas Lenslock y hemos podido comprobar que, en contra de lo que se venía afirmando, su utilización no representa apenas ninguna complicación.

Al acabar de cargar el programa, aparecen en la pantalla unos extraños símbolos que, aparentemente, no tienen ningún sentido. Al poner la lente sobre la pantalla, y realizando unos pequeños ajustes de ampliación o reducción de dichos símbolos (dependiendo del tamaño del monitor), podremos leer el mensaje O.K. Al pulsar Enter, y si no cambiamos la posición de la lente, aparecerá un código compuesto de letras y números que será el que tendremos que teclear para tener acceso al programa.

Como podréis apreciar, el manejo es, aunque algo incómodo relativamente sencillo, y los únicos inconvenientes que presenta están dirigidos hacia aquellos que deseen realizar una copia pirata del juego, por lo que podemos augurar un gran éxito a este ingenioso invento y seguramente en los próximos meses empezaremos a ver un buen número de programas que adopten esta medida de protección.



SORTEO ESPECIAL YOUR COMPUTER «GANA UN 128K»

En el sorteo especial de Your Computer, celebrado el día 5 de febrero entre todos los usuarios que nos mandaron sus datos, resultó agraciado con un Amstrad 128K.

José Manuel García Lagos. Pintos (Madrid).

Nuestra más sincera enhorabuena a este estudiante de 14 años, cuyo principal hobby es pasarse todo el tiempo que puede delante de su ordenador.

CLUB PARA USUARIOS DEL QL

En las últimas fechas se ha llevado a cabo la formación del C.E.I.U.Q.L. (Club Español Independiente de Usuarios del QL), el cual tiene la intención de servir de medio de unión, información y comunicación entre todos los usuarios de este ordenador, tanto españoles como hispanohablantes.

Este Club publicará un boletín con carácter mensual, en el que tendrán cabida todas las preguntas, dudas y problemas que los socios planteen junto con las soluciones a dichos interrogantes, así como todo tipo de colaboraciones referentes a comentarios o artículos sobre todos los temas relacionados con el QL.



Otros de los aspectos más importantes del club es la formación de una librería de programas de cartuchos, en la cual se almacenarán los programas que los socios envíen y a la que tendrán acceso todos aquellos miembros que así lo deseen.

La couta de inscripción es de 2.500 ptas. y desde el momento del abono se podrán solicitar los programas de la biblioteca, se recibirá mensualmente todas las revistas del club y se podrán enviar las colaboraciones o programas para su publicación.

Para mayor información ponerse en contacto con D. Serafín Olcoz. Baltasar Gracián, 21. 50005 Zaragoza.

YABBA DABBA DOO

LOS PICAPIEDRA AL ORDENADOR

La familia de los Picapiedra al completo, Pedro, Wilma y Pebble, hacen una nueva aparición en las pantallas, pero en esta ocasión no de los televisores sino de los ordenadores.

El programa tiene por título Yabba Dabba Do, y nos llega de la mano de Quicksilva, creadora de juegos como Ant Attack, Glass o Ruppert, en las diferentes versiones para Spectrum y Commodore 64.

Los archipopulares personajes de los comics y de la televisión, nos proponen una divertida arcade-aventura que nos transporta a su tan especial y simpática edad de piedra y en la cual tendremos la misión de ayudar a Pedro a solventar un pequeño problema que tiene entre manos.

Se trata de que para reconquistar el amor de su adorada Wilma, Pablo se ve obligado a construirle un lindo chalet en las afueras de la pietrópolis de Bedrock, pero debido a que las compañías constructoras todavía no funcionaban demasiado bien por aquellas épocas, deberá ir construyéndola piedra a piedra, evitando, además, los continuos ataques de los monstruos propios del lugar.

En breve podremos comprobar si los programadores de Quicksilva han hecho honor a tan ilustre personaje y han llevado a cabo un juego digno del simpático Pedro Picapiedra.



TRUCOS

MENSAJES ULTRASECRETOS

Este sencillo programa de mensajes secretos que nos envía S.B.E., nos será de mucha utilidad para dejar a buen seguro (ya que descifrarlos será poco más que imposible para cualquier «entrometido») nuestros documentos ultrasecretos. El programa, por otro lado, lo mismo sirve para poner en clave nuestros mensajes como para descifrarlos.

```
  10 INPUT a$: LET h=LEN a$
  20 FOR k=1 TO h
  30 LET x=CODE a$(k)
  40 IF (x-65)*(x-90)<=0 THEN LE
T x=155-x: GO TO 50
  45 IF (x-97)*(x-122)<=0 THEN L
ET x=219-x
  50 PRINT CHR$ x;
  60 NEXT k
  70 PRINT " ";: GO TO 10
```



LA SIRENA

Esta rutina en código máquina que os ofrecemos a continuación y que debemos a José Ignacio Linares, simula una sirena de coche, para lo cual aumenta y disminuye la frecuencia alternativamente 5 veces.

```
  10 FOR n=23300 TO 23349
  20 READ a: POKE n,a: NEXT n
  30 DATA 62,5,243,1,20,0,33,44,
1,17,7,0,205,42,91,43,16,250
  40 DATA 1,20,0,33,90,0,17,5,0,
205,42,91,35,16,250,61,32,222
  50 DATA 251,201,245,197,213,22
9,205,181,3,225,209,193,241,201
  60 RANDOMIZE USR 23300
```

AUTODESTRUCCION

Este truco está destinado a provocar la autodestrucción de nuestro ultrasecreto programa si alguien consigue traspasar sus defensas.

Consite en colocar la linea:

```
XXX IF PEEK 23625<>XXX
THEN PRINT USR 0
```

siendo XXX una línea menor de 255, la entraremos cuando el programa esté terminado, de manera que el cursor se quede allí.

Si alguien rompe el programa, lo más seguro es que haga un LIST, con lo que cambiará la dirección del cursor. A partir de aquí, todo intento de ejecutar el programa conllevará su autodestrucción en tanto no se restaure el cursor a su línea original o se elimine la línea XXX. Podemos colocar tantas trampas como queramos con la condición de que XXX sea siempre el mismo dentro de la sentencia y el cursor esté en la línea XXX.

Este truco se lo debemos a Xavier Casademont.

CANIBALES

Alberto TOLEDO

Spectrum 48 K

A veces, y para romper la norma, las cosas no ocurren como deben ocurrir, o al menos, como nos tienen acostumbrados a que ocurran; pues bien, este es el caso de la historia en la que os vais a sumergir si tecleáis este programa.

En plena selva brasileña, una orde de misioneros (sí, habéis entendido bien) se dispone, lanza en ristre, a asaltar a cuantos caníbales se interpongan en su camino para saborear su exquisita, aseguran, carne oscura.

No es que una extraña locura haya hecho mella en ellos, tampoco se trata de ningún malentendido. La razón es mucho más sencilla: una extraña plaga ha debastado la zona donde se ubica la misión, y cualquier tipo de alimento ha quedado fulminado, por lo que el hambre se ha convertido en el estado normal de los pobres misioneros.

Ante esta situación, el síndrome de abstinencia les ha llevado a echar mano a la más fresca carne disponible que tienen, la de los salvajes caníbales que antaño perseguían a los misioneros.

Así pues, como entre caníbales anda el juego, no olvidaros que a quien tenéis que ayudar es a los misioneros y lograr meter el mayor número de nativos en la olla.

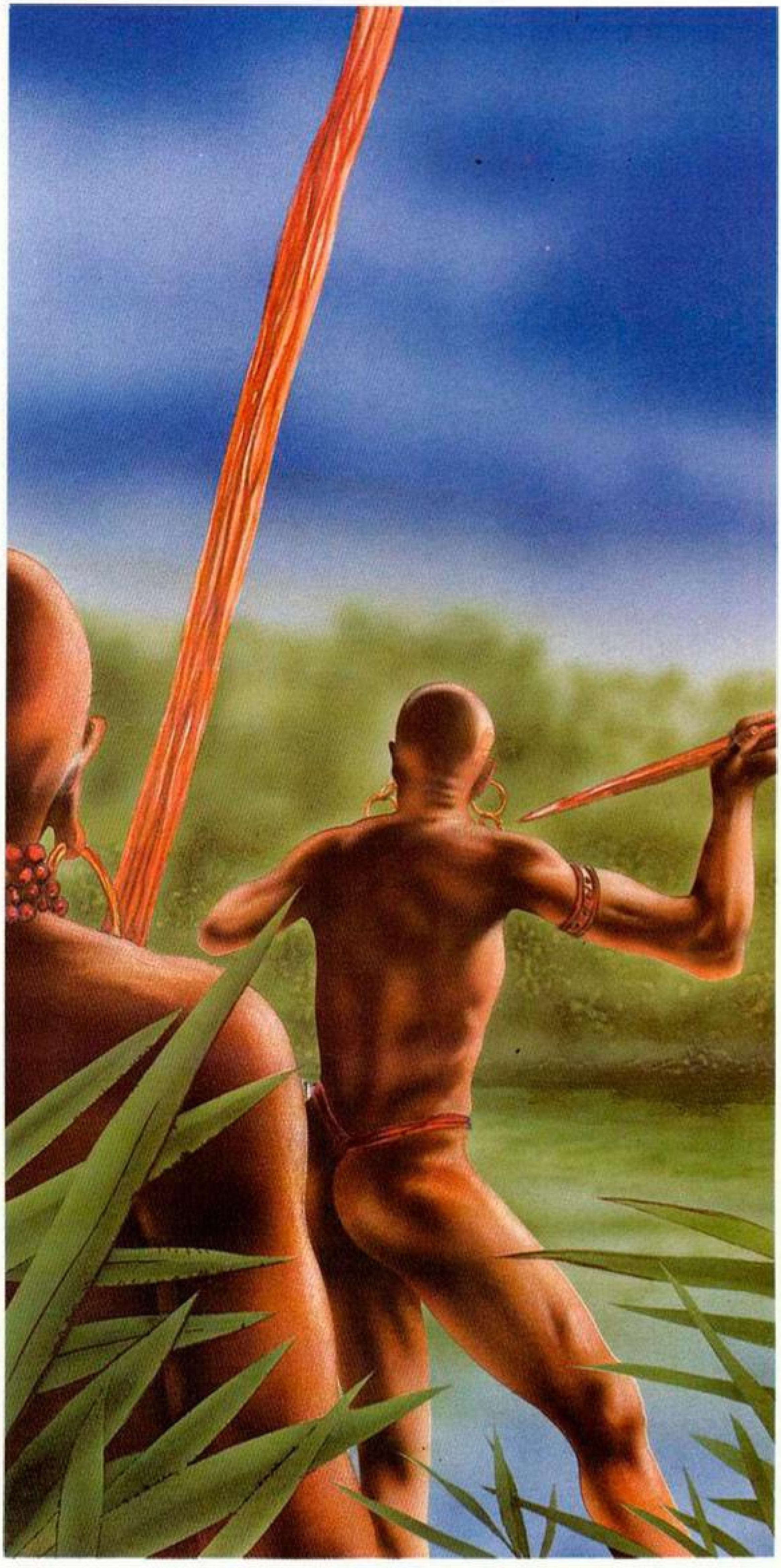
Los controles son:

O — Izquierda.
P — Derecha.
Q — Arriba.
A — Abajo.

¡Buen provecho...!

TODAS LAS LETRAS MAYUSCULAS SUBRAYADAS DEBERAN TECLEARSE EN MODO GRAFICO.

```
  50 BORDER 7: PAPER 7: INK 0: C
LS
 100 GO SUB 9000
 200 LET x=2: LET y=7: LET e=1:
LET ene=1: LET vi=4: LET x4=14:
```



```
LET y4=10: LET mi=0: LET cn=0: L
ET pant=1: LET ca=0: LET num=0:
LET h=1
 205 LET s$="EGFUGUFEFGEU"
 210 DIM i(3): DIM e$(2): DIM m$
(2): LET m$(1)="B": LET m$(2)="C
"
 212 FOR m=1 TO 3: LET i(m)=INT
(RND*48)+2: NEXT m
 220 GO SUB 3000
1000 REM MOVIMIENTO
1100 IF INKEY$="q" AND ATTR (y-1
,x)=56 THEN LET y=y-1: PRINT AT
y,x;m$(h);AT y+1,x;" ": IF y=1 T
HEN LET pant=pant-10: LET y=18:
GO SUB 3000
1120 IF INKEY$="a" AND ATTR (y+1
,x)=56 THEN LET y=y+1: PRINT AT
y,x;m$(h);AT y-1,x;" ": IF y=19
THEN LET pant=pant+10: LET y=2:
GO SUB 3000
1140 IF INKEY$="o" AND ATTR (y,x
-1)=56 THEN LET x=x-1: LET h=2:
PRINT AT y,x;m$(h);AT y,x+1;" ":
 IF x=1 THEN LET pant=pant-1: LE
T x=28: GO SUB 3000
1160 IF INKEY$="p" AND ATTR (y,x
+1)=56 THEN LET x=x+1: LET h=1:
PRINT AT y,x;m$(h);AT y,x-1;" ":
 IF x=29 THEN LET pant=pant+1: L
ET x=2: GO SUB 3000
1170 IF s>2 AND pant<>1 THEN GO
SUB 2000
1180 IF ca>0 AND pant=1 THEN IF
x=9 AND y<6 AND cn=1 THEN PRINT
AT 3,3+ca;"O": LET mi=mi+1: BEEP
 .1,0: BEEP .1,10: BEEP .1,-10:
LET cn=0: IF mi=3 THEN GO TO 400
0
1190 IF pant=i(mi+1) AND cn=0 TH
EN GO SUB 2500
1300 GO TO 1100
1999 REM ANIMALES
2005 IF ene=1 THEN LET ene=-ene:
 GO TO 2030
2010 LET ene=-ene: IF y>y1 AND A
TTR (y1+1,x1)=56 THEN LET y1=y1+
1: PRINT AT y1-1,x1;" ": GO TO 2
050
2020 IF y<y1 AND ATTR (y1-1,x1)=
56 THEN LET y1=y1-1: PRINT AT y1
+1,x1;" ": GO TO 2050
2030 IF x<x1 AND ATTR (y1,x1-1)=
56 THEN LET x1=x1-1: LET e=1: PR
INT AT y1,x1+1;" ": GO TO 2050
2040 IF x>x1 AND ATTR (y1,x1+1)=
56 THEN LET x1=x1+1: LET e=2: PR
INT AT y1,x1-1;" ": GO TO 2050
2045 IF ene=-1 THEN GO TO 2010
2050 PRINT AT y1,x1; INK co;e$(e
)
2055 IF y1=y AND x1=x THEN BEEP
.1,20: BEEP .1,-10: PRINT AT 10,
10; INK 1; FLASH 1;"TE COMIERON"
: FOR a=1 TO 100: NEXT a: LET vi
=vi-1: GO SUB 3000
2060 IF vi=0 THEN CLS : PRINT AT
 9,10; INK 1; FLASH 1;"HAS FRACA
SADO";AT 11,9;"NO TIENES VIDAS":
 GO TO 9900
2080 RETURN
2500 REM CANIBALES
2510 IF x>x4 AND ATTR (y4,x4-1)=
56 THEN LET x4=x4-1: PRINT AT y4
,x4+1;" ";AT y4,x4;"J": IF x4=1
THEN LET i(ca+1)=i(ca+1)-1: PRIN
T AT y4,x4;" ": LET x4=26: GO TO
 2550
2520 IF x<x4 AND ATTR (y4,x4+1)=
56 THEN LET x4=x4+1: PRINT AT y4
,x4-1;" ";AT y4,x4;"J": IF x4=29
 THEN LET i(ca+1)=i(ca+1)+1: PRI
NT AT y4,x4;" ": LET x4=4: GO TO
 2550
2525 IF ATTR (y4,x4+1)=56 AND AT
TR (y4,x4-1)=56 AND x4>2 AND x4<
28 THEN GO TO 2550
2530 IF y<y4 AND ATTR (y4+1,x4)=
56 THEN LET y4=y4+1: PRINT AT y4
-1,x4;" ";AT y4,x4;"J": IF y4=19
 THEN LET i(ca+1)=i(ca+1)+10: PR
INT AT y4,x4;" ": LET y4=4: GO T
O 2550
2540 IF y>y4 AND ATTR (y4-1,x4)=
56 THEN LET y4=y4-1: PRINT AT y4
+1,x4;" ";AT y4,x4;"J": IF y4=1
THEN LET i(ca+1)=i(ca+1)-10: PRI
NT AT y4,x4;" ": LET y4=16: GO T
O 2550
2550 IF x=x4 AND y=y4 THEN BEEP
.1,0: BEEP .1,-5: LET ca=ca+1: L
ET i(ca)=0: LET x4=14: LET y4=10
: LET cn=1: PRINT AT 21,10; INK
1; FLASH 1;"LO HAS CAZADO"
2560 RETURN
2999 REM JUNGLA
3000 CLS : LET y1=10: LET x1=14
3005 PRINT #0;"VIDAS: ";vi,"CANIB
ALES: ";mi
3007 IF vi=0 THEN CLS
3010 LET c$=STR$ pant: IF c$(LEN
 c$)="1" THEN FOR a=1 TO 19: PRI
NT INK 4;AT a,1;"C": NEXT a
3020 IF c$(LEN c$)="0" THEN FOR
a=1 TO 19: PRINT INK 4;AT a,29;"
C": NEXT a
3030 IF pant>40 THEN PRINT INK 4
;AT 19,1;"EFEFGEFGEEFGGEFEFEGGGG
EFEFGGE"
3040 IF pant<=10 THEN PRINT INK
4;AT 1,1;"EEEFGGGEFGEFEFGGEFEGGE
EFEGGFG"
3050 LET x3=INT (RND*20)+2: LET
y3=INT (RND*17)+2: LET s=INT (RN
D*6)+1: PRINT AT y3,x3; INK 4;s$
(3 TO 3+s): LET num=num+1
3055 IF num<6 THEN GO TO 3050
3057 LET num=0
3060 LET x2=INT (RND*27)+2: LET
y2=INT (RND*16)+2: IF ATTR (y2,x
2)=56 AND ATTR (y2+1,x2)=56 THEN
 PRINT AT y2,x2; INK 4;"C";AT y2
+1,x2; INK 3;"D": LET num=num+1
3065 IF num<5 THEN GO TO 3060
3067 LET num=0
3069 IF s=3 THEN LET e$(1)="S":
LET e$(2)="T"
3070 IF s=4 THEN LET e$(1)="M":
LET e$(2)="N"
3072 IF s=5 THEN LET e$(1)="M":
LET e$(2)="I"
3073 IF s=6 THEN LET e$(1)="Q":
LET e$(2)="R"
3074 LET co=INT (RND*3)+1
3075 IF pant=1 THEN FOR a=2 TO 6
: PRINT AT a,2; INK 4;"      ":
 NEXT a: PRINT INK 1;AT 4,3;"K
 ": INK 2;AT 5,4;"PPP"; INK 4;A
T 6,2;"       "
3076 IF pant=1 THEN PRINT AT 7,2
;"     "
3077 IF pant=1 THEN FOR c=1 TO m
i: PRINT AT 3,3+c;"O": NEXT c
3079 IF ATTR (y1,x1)<>56 THEN LE
T x1=x1+1: GO TO 3079
3080 IF ATTR (y4,x4)<>56 THEN LE
T x4=x4+1: GO TO 3080
3095 IF ATTR (y,x)<>56 THEN LET
x=x+(x<27): GO TO 3095
3096 IF ATTR (y,x)=56 THEN GO TO
 3099
3097 LET x=x-1
3099 PRINT AT y,x;m$(h)
3100 RETURN
3999 REM FINAL
4000 FOR a=1 TO 20: BEEP .01,a:
BEEP .01,a+1: BEEP .01,a+2: BEEP
 .01,-a: NEXT a
4010 CLS : PRINT AT 2,5; FLASH 1
; INK 1;"LO HAS CONSEGUIDO!!"'''
'"HOY TE DARAS UN BUEN ATRACON"'
' INK 2; FLASH 0;"LOS CANIBALES
YA ESTAN LISTOS          PARA E
L PUCHERO"'''''; INK 4; FLASH 1;
"!!!!QUE TE LOS COMAS BIEN!!!!"
4020 PRINT AT 19,0; INK 1;"EFFEG
EFEGEFEGFEGGEFEGFUGFGUGEUFUG
     Pulsa una tecla        GEFGUU
FEFUEFEUFGFGFUFGEUFGUGFGEGE"
4030 PAUSE 0: GO TO 9900
9000 REM GRAFICOS
9010 FOR g=USR "a" TO USR "u"+7:
 READ da: POKE g,da: NEXT g
9020 DATA 24,126,24,26,60,88,20,
34,24,126,24,88,60,26,40,68,60,6
6,145,165,149,193,38,24,24,28,28
,24,56,24,60,126
9030 DATA 60,74,82,60,24,24,24,6
0,108,56,16,80,48,28,16,60,38,88
,62,184,156,91,60,60
9040 DATA 192,160,124,62,62,38,3
7,109,3,5,62,124,124,100,164,182
,24,152,191,219,155,152,164,164
9050 DATA 255,255,255,127,63,31,
15,3,255,255,255,254,252,248,240
,192
9060 DATA 96,224,64,64,72,84,85,
34,6,7,2,2,18,42,170,68
9070 DATA 189,165,189,153,126,60
,60,60,74,41,82,85,149,237,126,6
0
9080 DATA 0,4,14,206,223,127,18,
54,0,32,112,115,251,254,72,108,0
,2,5,193,255,62,81,81,0,64,160,1
31,255,124,138,138,153,219,189,1
53,126,24,24,60
9110 FLASH 1: PRINT INK 1;AT 6,7
;"SOPA DE CANIBALES"; INK 2;AT 1
0,7;" © ALBERTO TOLEDO"; INK 3;A
T 14,11;" 1 9 8 5 "
9120 FLASH 0: FOR t=-20 TO 30: B
EEP .01,t: BEEP .01,-t: BEEP .01
,t+1: BEEP .01,t+5: BEEP .01,t-1
0: NEXT t
9200 GO TO 200
9210 IF i$="n" THEN GO TO 200
9220 GO TO 9200
9500 RETURN
9900 FOR w=-10 TO 20: BEEP .01,w
: BEEP .01,-w: NEXT w
9910 INPUT "Otro juego (s/n)? ";
j$: IF j$="s" THEN BEEP 1,0: BEE
P 1,-10: GO TO 200
9920 IF j$="n" THEN FOR g=30 TO
-20 STEP -1: BEEP .01,g: NEXT g:
 STOP
9930 GO TO 9910
```

AHORA SÍ

puedes aprender a programar en basic de una vez por todas

¡Solicítalo antes de que se agote!
Hay un número limitado de ejemplares

DEJATE de complicados e incomprensibles sistemas de aprendizaje. Conoce de una vez por todas lo que es el Basic. Es más sencillo de lo que crees, porque ahora tienes algo que estabas esperando hace mucho tiempo: MICROBASIC, una edición corregida y revisada del famoso curso publicado por MICROHOBBY SEMANAL.

MICROBASIC es el libro que te enseñará a ser un experto en programación. Aunque hasta ahora sólo hayas utilizado tu Spectrum para jugar.

MICROBASIC te introducirá, paso a paso, en el Basic. Con ejemplos claros, sencillos y prácticos que irán adquiriendo complejidad según vayas aumentando tu nivel. Hasta llegar a dominarlo por completo.

Aprovecha esta oportunidad, porque ahora sí puedes llegar a conocer a fondo tu Spectrum. Ahora, por fin, a tu alcance el método más claro y completo de programación en Basic publicado hasta el momento.



Recorta o copia este cupón y envíalo a HOBBY PRESS, S. A. Apartado de Correos 232. Alcobendas (Madrid)

Nombre ______
Apellidos ______
Dirección ______
Localidad ______ Provincia ______
Código Postal ______ Edad ______ Teléfono ______

Deseo recibir en mi domicilio el libro MICROBASIC, al precio de 1.750 ptas. (IVA incluido). El importe lo pagaré:

- ☐ Mediante talón bancario adjunto a nombre de HOBBY PRESS, S. A.
- ☐ Mediante tarjeta de crédito ______
  Número de la tarjeta ______
  Fecha de caducidad de la tarjeta ______
- ☐ Mediante giro postal n.º ______
- ☐ Contra reembolso (supone 75 ptas. de gastos de envío)

Fecha y firma

¡NUEVO!

CAMELOT WARRIORS • Arcade • Dinamic

CAMELOT WARRIORS

UNA FANTASIA MEDIEVAL

Dinamic nos propone una emocionante aventura en la que mundos pasados y futuros no tienen fronteras, un alucinante encuentro en un lugar donde la magia y la fuerza de tu espada modifican a su antojo el destino de los hombres. ¿Te atrae la idea?

Camelot Warriors posiblemente sea el mejor juego que Dinamic ha realizado hasta el momento en su ya dilatada carrera. Esto es debido, no sólo a su buena presentación y excelentes gráficos, sino también a que, en contra de a lo que nos tenía acostumbrados en los últimos programas, la acción se desarrolla en muchos y muy variados escenarios y todo el juego se encuentra cargado de una gran diversidad de formas y colores.

El argumento de este singular arcade nos transforma en un aguerrido caballero de aspecto medieval, espada en ristre y enfundado en una resistente armadura, y nos encomienda la difícil misión de encontrar los cuatro objetos que, provenientes de alguna lejana época futura, han ido misteriosamente a aparecer en estos mundos medievales.

Los cuatro objetos a encontrar son: la Voz de otro mundo, el Espejo de la sabiduría, el Elixir de la vida y el Fuego que no quema. Cada uno de ellos resultarán muy sencillos de identificar, aunque por ética profesional no os vamos a desvelar qué es cada uno de ellos, aunque sí os podemos decir que os van a resultar muy familiares.

Para poder conseguir los objetos, deberemos ir enfrentándonos a los peligros propios de cada mundo: el Bosque, el Lago, las Grutas y finalmente, el Castillo de Cámelot. En ellos encontraremos un gran número de atacantes de la más diversa índole y naturaleza: enormes abejorros, seres amorfos, peces devoradores, horribles alimañas, arañas peludas... Para defendernos de ellos contamos con la inestimable ayuda de nuestra inseparable espada y con nuestras grandes dotes de atleta, ya que, gracias a nuestro potente salto, podremos esquivar a estos temibles enemigos o, en otras ocasiones, escalar por los intrincados riscos de las grutas o el bosque.

Cada vez que consigamos alguno de los cuatro mágicos objetos, deberemos entregarlos a los guardianes que se encargan de defender las fronteras de cada muro. Estos nos transportarán a través de una nueva dimensión, al fantástico lugar donde se nos permitirá continuar con nuestra afanosa búsqueda.

Estos personajes son: Aznaht, el druida del Bosque y al que deberemos entregar el Fuego que no quema; Kindo, rey del Lago a quien obsequiaremos el Espejo de la sabiduría; Azornic, dragón señor de las grutas, ante el cual tendremos que presentarnos con el Elixir de la vida y por último, Arturo, Rey de Camelot y con el que acabaremos la aventura presentándonos ante él con la Voz de otro mundo.

Como estaréis comprobando, Camelot Warriors es un juego en el que la fantasía y la imaginación lo inundan todo; pero afortunadamente, éstas no son sus únicas virtudes.

Otra característica que llama poderosamente la atención es el alto nivel de calidad gráfica con el que están realizadas todas y cada una de las numerosas pantallas que configuran el juego.

Esto, evidentemente, no representa ninguna novedad, pues Dinamic nos ha venido demostrando en sus últimos programas que el diseño es algo que sus programadores dominan a la perfección. Sin embargo, en este último juego podemos encontrar, además de una notable calidad, una notable variedad. Las pantallas son totalmente distintas entre sí, y cada vez que conseguimos pasar de una a otra, sobre todo cuando cambiamos de un mundo a otro, nos encontramos con la agradable sorpresa de que nos da la sensación de estar ante otro juego diferente, lo que nos incita a intentar llegar más lejos, casi con el único objetivo de poder admirar qué sorpresa nos aguarda en el siguiente escenario.

Camelot Warriors pertenece a ese tipo de juegos en los que se consigue un alto grado de adición simplemente por el hecho de tener la oportunidad de admirar sus gráficos. Aquí no se puede luchar por conseguir una puntuación más alta por el mero detalle de que carece de puntuación, y tampoco tendremos que esforzar-

Lo conseguí...

Si eres uno de los intrépidos guerreros que han conseguido superar las arduas pruebas a que ha sido sometido CAMELOT WARRIOR y has logrado destruir EL ESPEJO DE LA SABIDURIA, EL FUEGO QUE NO QUEMA, EL ELIXIR DE LA VIDA Y LA VOZ DEL OTRO MUNDO, habrás descubierto el secreto de CAMELOT WARRIORS.

Envíanoslo en un sobre, junto con las instrucciones que acompañan al juego original (impreso verde).

Los cien primeros acertantes, cuyas cartas recibamos y cumplan con estos requisitos (se mirará la fecha del matasellos), recibirán en su domicilio, COMPLETAMENTE GRATIS, una explendida camiseta diseñada en exclusiva para los ganadores. NO HAY NINGUNA OTRA FORMA DE CONSEGUIRLA.



P.D. Si deseas recuperar las instrucciones adjunta también un sobre FRANQUEADO con tu dirección.

nos por intentar lograr nuestro objetivo en el menor espacio de tiempo posible, ya que ni siquiera tiene un reloj que nos acose.

Pero tampoco debéis pensar que todo el encanto de Camelot Warriors reside en el aspecto visual del mismo. Además, es un juego bastante divertido y emocionante.

Los peligros que nos aguardan en cada pantalla son muy diversos, y, sin llegar a ser agobiantes, pues tampoco es que nos estén acosando constantemente y que haya demasiados enemigos, se ha conseguido que tengamos que permanecer casi todo el tiempo en tensión y que incluso en muchas ocasiones tengamos que elaborarnos alguna que otra pequeña táctica para poder sortear los obstáculos que se nos presenten. Esto se consigue gracias a las variadas acciones que podemos realizar y verdaderamente resulta muy entretenido ver a nuestro valeroso warrior de la corte de Camelot demostrando sus increíbles habilidades. Va-

EL MUNDO DE CAMELOT WARRIORS



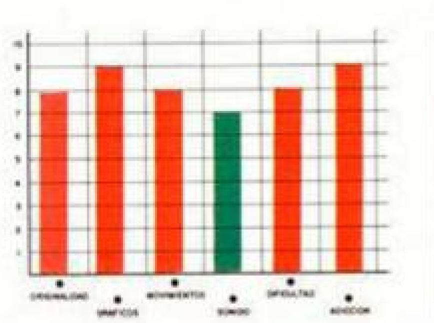


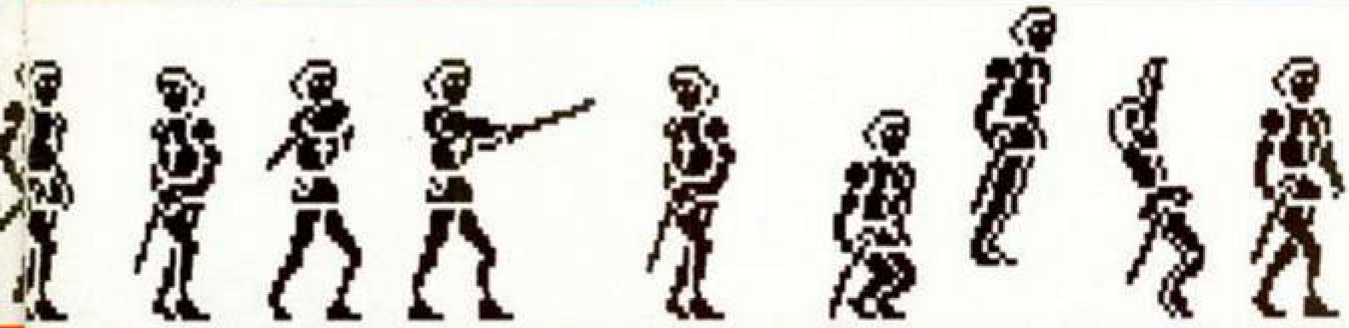




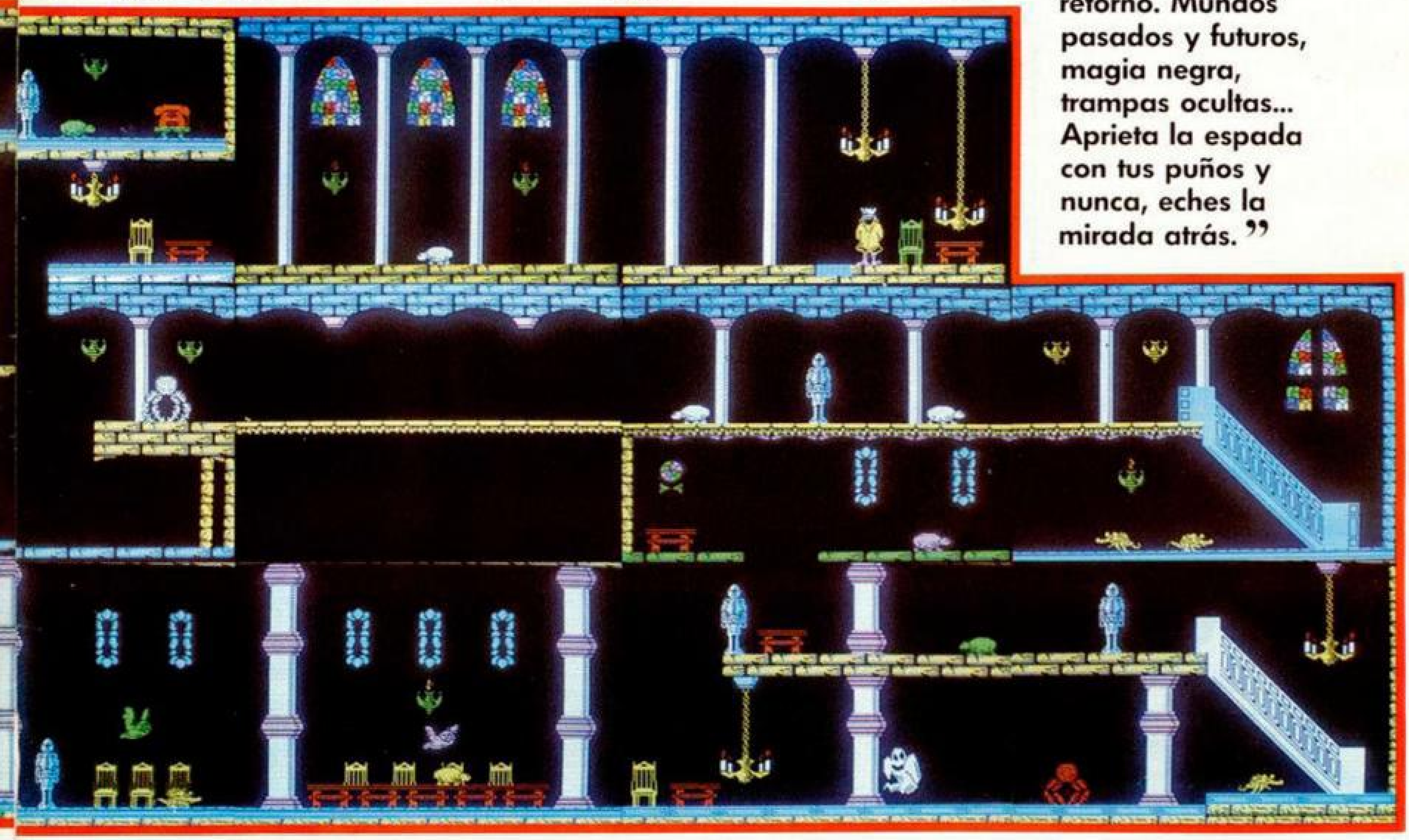
mos, que seguro que disfrutaréis de lo lindo observando cómo el guerrero salta, escala, desenfunda rápidamente su espada, mata bichos, se transforma en rana, coge objetos y realiza todo tipo de vistosos movimientos.

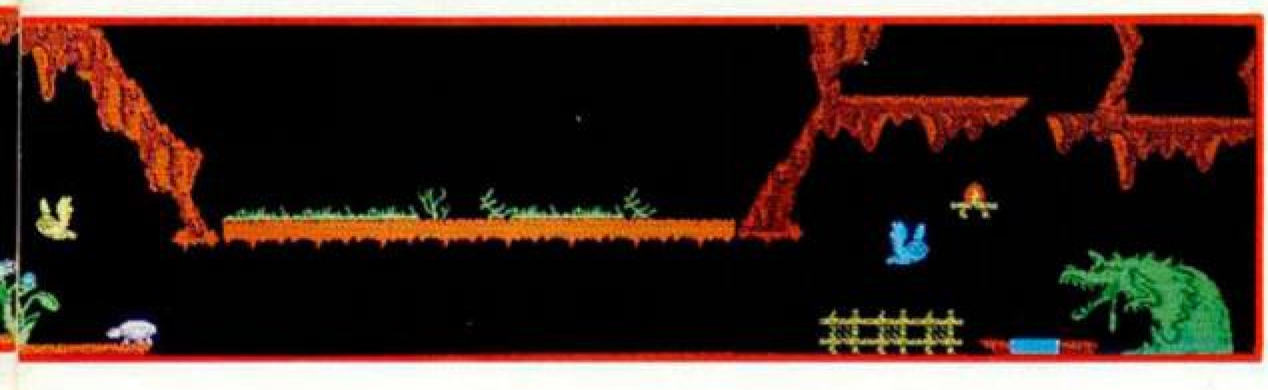
Así pues, ¿qué más podemos pedir de Camelot Warriors? ¿Originalidad? La tiene. ¿Bonitos gráficos? Los tiene. ¿Buenos movimientos? También, y en cuanto a la diversión y el «pique» están totalmente asegurados... Entonces ¿qué ocurre? Pues ocurre que nos encontramos ante un excelente programa. Nada más.





“Si osas franquear la puerta del misterio, olvida todo lo que conoces porque te internarás en un viaje sin retorno. Mundos pasados y futuros, magia negra, trampas ocultas... Aprieta la espada con tus puños y nunca, eches la mirada atrás.”





• **El bosque:** Escenario natural en que te esperan numerosos enemigos. Consigue que el hechizo del Druida te transforme en un viscoso batracio.

• **El lago:** Peces asesinos, medusas eléctricas y un ambiente terrorífico.

•**Las grutas:** Sumergido en las entrañas de la tierra, el dragón Azornic, señor de las grutas, espera el elixir de la vida.

•**El castillo de Camelot:** El Rey Arturo tiene en su poder todo el secreto de esta apasionante aventura.

¡NUEVO!

DESERT RATS • War-Game • Cases Computers Simulations

LA DERROTA DEL AFRIKA CORPS

Para deleite y disfrute de los viciosos de los juegos de guerra, y para iniciar la adicción de aquellos que aún no lo son, Cases Computers Simulations ha creado este estupendo Desert Rats.

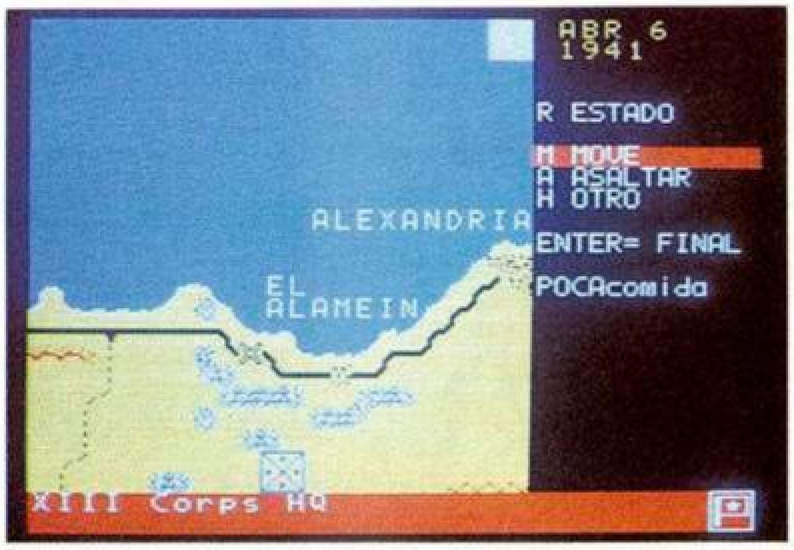
En esta ocasión tendremos que ponernos ligeros de ropa y llenar hasta arriba nuestras cantimploras, porque nos vamos nada menos que al crudo y árido desierto Norteafricano. Allí tendremos la oportunidad de revivir, a través de nuestro ordenador, los acontecimientos que tuvieron lugar en esta zona desde la llegada de los Afrika Corps en 1941, hasta su derrota en la batalla de Alamein y su consiguiente retirada del territorio de Libia, allá por los finales del año 1942.

En Desert Rats pueden competir dos jugadores entre sí, o bien uno sólo frente al ordenador, y se ofrece la posibilidad de elegir entre representar el papel de las tropas británicas (Aliados) o las italo-germanas (Eje).

Por otra parte, también se nos permite las opciones de llevar a cabo la campaña completa o la de ir afrontando por separado cada una de las cinco batallas más importantes que tuvieron lugar durante el desarrollo de la misma: la entrada de Rommel, la operación Hacha de Guerra, la Operación Cruzada, la batalla de Gazala o El Alamein.

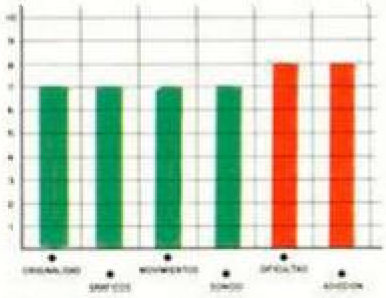
Cada una de ellas posee, lógicamente, sus circunstancias específicas, y los objetivos a cumplir dependen del número de turnos disponibles, de la cantidad de tropas y de las peculiaridades de la zona y territorios a ocupar. Cuando hayamos conseguido salir victoriosos de cualquiera de estas confrontaciones, podremos continuar con la siguiente misión. El hecho de poder asumir cada batalla por separado le da una gran variedad al juego, ya que la estrategia difiere considerablemente al intentar llevar a cabo toda la misión al completo.

Otra característica destacable de este magnífico war-game es la gran cantidad de unidades que forman parte de las batallas: tanques de todo tipo, unidades de infantería, grupos de apoyo, infantería motorizada, artillería, etc..., que hacen de Desert Rats un juego realmente completo y con el que tendrás que demostrar al máximo tus habilidades y dotes de estratega.

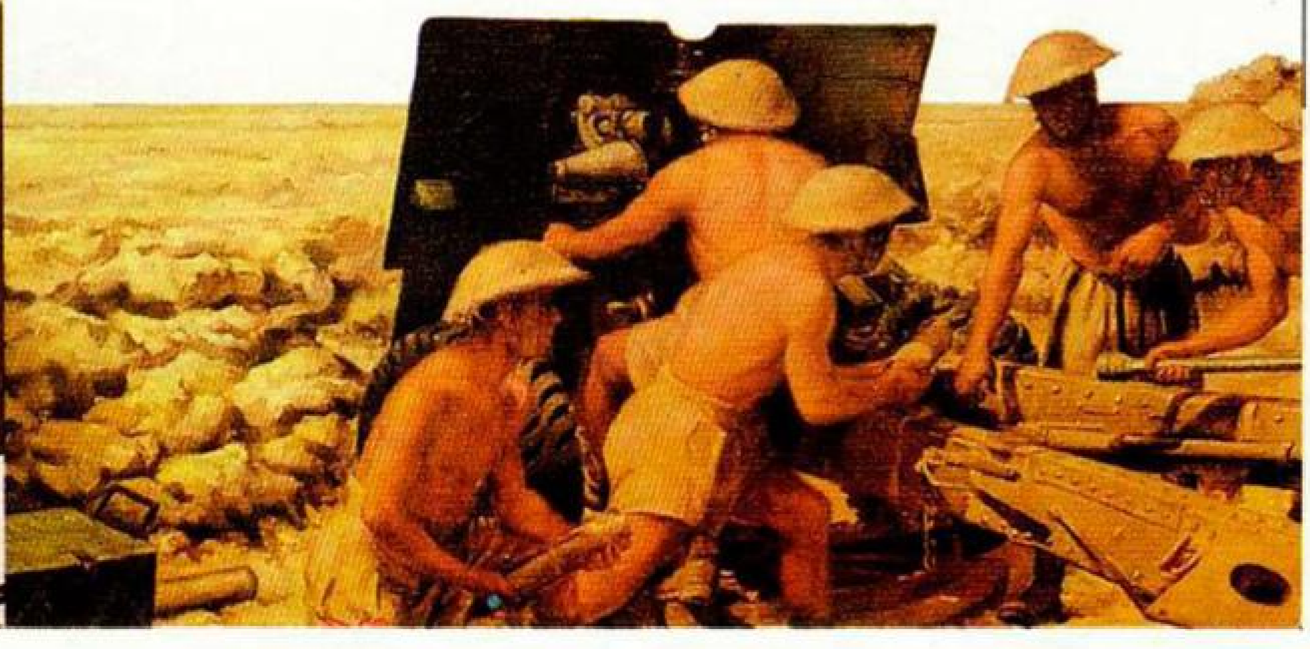


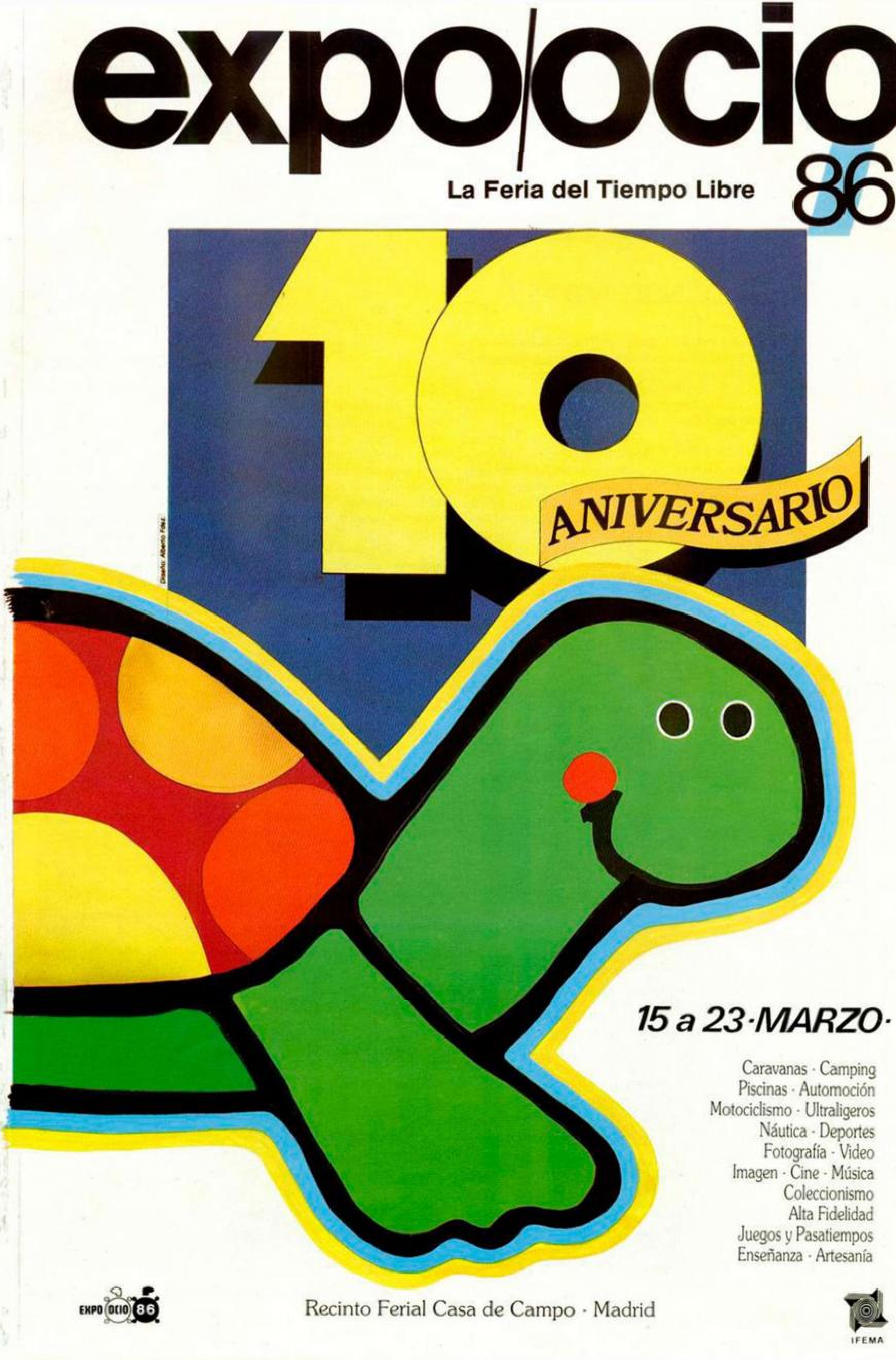
Aquellos que hayan tenido la oportunidad de jugar con otro war-game llamado Arhnem, no tardarán en reconocer el mismo estilo de juego. Esto no es producto de la casualidad, ya que R. T. Smith es el creador de ambos, pero en este nuevo programa ha añadido una serie de innovaciones y mejoras en el sistema y desarrollo del juego, y ha conseguido con Desert Rats mejorar sus anteriores obras.

Un juego que recomendamos muy especialmente a adictos a los war-games y a todos aquellos que aún piensan que los juegos de ordenador son cosa de niños.

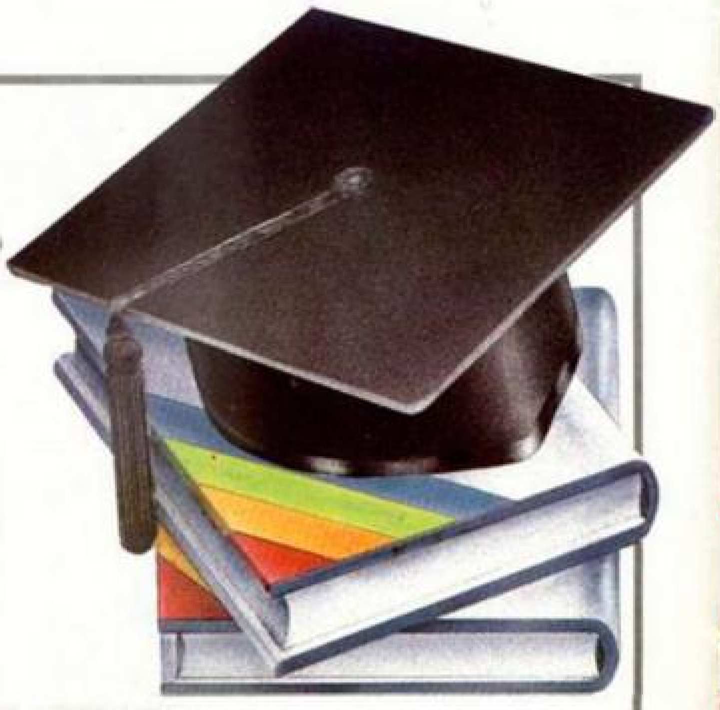







expo/ocio
86
La Feria del Tiempo Libre
10
ANIVERSARIO
15 a 23·MARZO·
Caravanas · Camping
Piscinas · Automoción
Motociclismo · Ultraligeros
Náutica · Deportes
Fotografía · Video
Imagen · Cine · Música
Coleccionismo
Alta Fidelidad
Juegos y Pasatiempos
Enseñanza · Artesanía
EXPO OCIO 86
Recinto Ferial Casa de Campo · Madrid
IFEMA

Profesor particular



Arturo LOBO y J. J. LEON

INTEGRALES INDEFINIDAS I

Por cuestiones de espacio, el programa de esta semana hemos tenido que dividirlo en dos. Ninguna de las partes puede funcionar sin la otra, sin embargo, hemos procurado que esta primera tuviera cierta autonomía y no tuvierais que esperar a la semana que viene para correr el programa.

El programa en su conjunto es muy potente en lo que se refiere al cálculo de integrales definidas e indefinidas. No queremos que nadie crea que calcula la primitiva de una función, lo que hace es definir una función FNi(x) que es una aproximación muy buena de la integral pedida.

Esta primera parte del programa sólo puede hallar intengrales definidas de una función, aunque lo hace con gran precisión y rapidez. A pesar de que en la pantalla aparece un menú con cuatro opciones, sólo la de integrales definidas funciona. Las otras tres son: integrales indefinidas, dibujo de la función, y dibujo de la función integral.

Para poder utilizar estas dos últimas opciones es necesario dar MERGE de este programa completo sobre el de «Representación Gráfica» que os ofrecimos hace unas semanas y que permite dibujar una función con una escala graduada horizontal y vertical para la perfecta visualización de la función.

Algunos lectores nos han escrito diciendo que tienen problemas a la hora de teclear la línea 9999 que aparece en éste y en otros programas anteriores, el problema consiste en introducir la cadena ":RETURN: REM" ya que dentro de las comillas no aparece el RETURN ni el REM, lo que se hace es teclear :RETURN: REM sin comillas y después poner las comillas a los lados. También tenemos que avisaros que es muy importante que esa línea esté escrita literalmente como viene en el listado, sin omitir ni incluir nada.

Os recomendamos que provéis el programa con funciones de las que ya conozcáis su primitiva y comparéis los resultados exactos y el que os da el programa. Esto lo podéis hacer tanto en integrales definidas como en indefinidas, y comparar las distintas aproximaciones.

El de integrales definidas pide un número llamado precisión, que también hace crecer el tiempo de ejecución que no siempre es conveniente que sea grande, si la función es sencilla o el intervalo de integración pequeño es mucho más rápido, elegir un número pequeño y el resultado sigue siendo fiable.

Conviene recordar que hay que tener cuidado al introducir las funciones en el ordenador. Por una parte, el programa la admitirá esté bien o mal escrita aunque luego dará error. Y por otra parte, el Spectrum no eleva números negativos a otros por lo que los polinomios y las funciones potenciales no pueden tener argumentos negativos, o dará error.

TODAS LAS LETRAS MAYUSCULAS SUBRAYADAS DEBERAN TECLEARSE EN MODO GRAFICO

```
3799 GO TO 7500
3800 LET menu1=3: LET y0=0
3810 DIM v(21,2): LET dimi=21: L
ET menu2=2
3815 GO SUB 4000
3820 DATA "         MENU ","1- INT
EGRAL DEFINIDA","2- INTEGRAL IND
EFINIDA","3- DIBUJAR FUNCION","4
- DIBUJAR INTEGRAL"
3830 CLS : RESTORE 3820: FOR I=1
 TO 5: READ z$: BEEP 1/(i↑2),i↑2
: PRINT AT 3+2*i,4;z$: NEXT i
3840 LET paw=0: LET mun=CODE INK
EY$: IF (mun>48)*(mun<53) THEN L
ET paw=10: CLS : GO SUB (6120*(m
un=49)+4010*(mun=50)+4500*(mun=5
1)+4525*(mun=52))
3850 GO TO (3840-paw)
4000 LET tiempo=120: LET inc=.1:
 LET grado=10
4002 BORDER 6: PAPER 6: INK 0: C
LS
4005 LET v$="": FOR w=1 TO grado
+1: LET v$=v$+"k(n,"+STR$ w+")*x
*(": NEXT w
4007 LET v$=v$+"k(n,"+STR$ (grad
o+2)+")"
4008 FOR w=1 TO grado+1: LET v$=
v$+")": NEXT w
4009 RETURN
4010 INPUT "METE f(x)=", LINE j$
: INPUT "Origen de integracion "
;"x0=";x0: INPUT "EXTREMO DEL IN
TERVALO DE""INTEGRACION   x1= ";
x1
4015 INPUT "MAXIMO ERROR DE INTE
GRACION =";ERROR
4020 GO SUB 5000
4023 LET gre=grado+(ult+5-grado)
*(ult+5<grado)
4024 DIM k(5,12)
4025 DIM x(5): LET derm=0: FOR p
=1 TO 5: LET x(p)=x0+(p-1)/4*(x1
-x0): GO SUB 5050: LET put=ABS k
(p,gre+2): LET derm=(derm+put+AB
S (derm-put))/2: NEXT p
4035 LET delt=(2*error/(derm+1e-
27))↑(1/(gre+1))
4040 LET delt=(delt+(x1-x0)/20+A
BS (delt-(x1-x0)/20))/2
4050 LET puntos=INT ((x1-x0)/del
t)+1
4060 DIM x(puntos): DIM k(puntos
,grado+2)
4070 FOR i=1 TO puntos: LET x(i)
=x0+delt*(i-1): NEXT i
4080 FOR p=1 TO puntos: LET x=x(
p)
4082 FOR w=1 TO ult+1: LET o=w+1
: LET i$="k(p,o)=VAL "+CHR$ (105
+w)+"$": GO SUB 9999: LET k(p,w+
1)=k(p,w+1)/FN j(w): NEXT w
4090 GO SUB 5050: NEXT p
4100 LET clx=0: FOR p=1 TO punto
s: LET k(p,1)=clx-FN i(x(p)-(del
t/(2+1e-5))*(p>1)): LET clx=FN i
(x(p)+delt/(2+1e-5))
4110 NEXT p
4120 PRINT AT 3,0;"LA FUNCION IN
TEGRA ,FN i(x),     ESTA HALLADA
VUELVO AL MENU"
4125 PAUSE 300: RETURN
4500 LET q$=j$+"-y": GO SUB 7002
4510 RETURN
4525 CLS : LET q$="FN i(x)-y": G
O SUB 7002: RETURN
4994 DEF FN i(x)=FN g(x-x(INT ((
x-x0)/delt+1.5)),INT ((x-x0)/del
t+1.5))
4995 DEF FN g(x,n)=VAL v$
4996 DEF FN k(x,n)=ABS x↑n*(SGN
(2*INT ABS (n/2)-ABS n+1e-20)*(x
<0)+(x>=0))
4997 DEF FN j(x)=(x<=1)+2*(x=2)+
6*(x=3)+24*(x=4)+120*(x=5)+720*(
x=6)+5040*(x=7)+40320*(x=8)+3628
80*(x=9)+3628800*(x=10)+39916800
*(x=11)
4998 DEF FN h(x)=VAL j$
4999 DEF FN d(x)=VAL e$
5000 GO SUB 5049: LET time0=time
: LET e$=j$: FOR w=2 TO grado+1:
 LET i$="x": GO SUB 8000: LET i$
=CHR$ (105+w)+"$=e$": GO SUB 999
9
5005 GO SUB 5049: IF time-time0<
tiempo THEN NEXT w: LET w=w-1
5010 LET ultima=w-1: LET ult=ult
ima
5020 RETURN
5040 LET fac=1: FOR i=1 TO j: LE
T fac=fac*i: NEXT i: RETURN
5049 LET time=(65536*PEEK 23674+
256*PEEK 23673+PEEK 23672)/50: R
ETURN
5050 LET dzj=2*(grado-ultima)+1:
 DIM z((dzj+1)/2,dzj): FOR j=1 T
O dzj: LET z(1,j)=FN d(x(p)+(j-(
dzj+1)/2)*inc): NEXT j
5060 GO SUB 5100: FOR j=ultima+1
 TO grado+1: LET k(p,j+1)=z(j-ul
tima,(dzj+1)/2)/FN j(j)*(j<ult+7
): NEXT j: RETURN
5100 FOR i=2 TO (dzj+1)/1: FOR j
=i TO dzj-i+1: LET z(i,j)=(z(i-1
,j+1)-z(i-1,j-1))/2/inc: NEXT j:
 NEXT i: RETURN
6120 INPUT "Mete la funcion f(x)
=", LINE w$
6130 INPUT "LIMITE INFERIOR = ";
asd
6140 INPUT "Limite superior =";b
sd
6160 INPUT "Precision (3 a 9)=";
jsd
6175 LET dsd=(bsd-asd)/2↑jsd
6180 LET x=asd: LET isd=VAL w$
6200 FOR x=asd+2*dsd TO bsd STEP
 2*dsd: LET jsd=VAL w$: LET isd=
isd+2*jsd
6210 NEXT x
6220 LET isd=isd-jsd
6240 FOR x=asd+dsd TO bsd STEP 2
*dsd: LET jsd=VAL w$: LET isd=is
d+4*jsd
6245 NEXT x
6250 LET isd=isd*dsd/3
6260 PRINT AT 5,0;"La integral p
edida vale..";"          ";isd'"" Pu
lsa una tecla para MENU"
6270 PAUSE 0: RETURN
7003 GO TO 7025
7028 LET pasa=0: GO TO 7063
7120 INPUT "PULSA ENTER PARA CON
TINUAR"; LINE B$: RETURN
7595 GO TO 3800
8350 LET a$=c$: RETURN
9999 LET args=PEEK 23627+256*PEE
K 23628: FOR i=1 TO (LEN i$+(19-
LEN i$)*(LEN i$>19)): POKE (args
-25+i),CODE (i$(i)): NEXT i: FOR
 j=1 TO 4: POKE (args-25+i+j-1),
CODE ":RETURN: REM ":(j): NEXT
j: LET k(p,o)=VAL t$: RETURN : R
EM : REM VAL b$: RETURN : REM
```

mido el «XOR A» que había en la línea 420 y, en vez de hacer «LD (HL),A» hacemos «LD (HL),L». Hemos simplificado el listado (este tiene dos líneas menos) y, además, funciona más deprisa. Aún podría simplificarse más; puesto que todos los octetos menos significativos son «ceros», podríamos almacenar en la tabla sólo los más significativos y cargar «L» y «E» con «0». En su día no se hizo así porque se pretendía que el ejemplo ilustrara, también, la forma de utilizar una tabla de «offset».

La rutina «CLS3» del capítulo 7 quedaría:

```
100         ORG  60000
110         LD   A,(23681)
120         AND  A
130         JR   Z,ERROR
140         CP   4
150         JR   C,CLS3
160 ERROR   RST  8
170         DEFB #0A
180 CLS3    DEC  A
190         ADD  A,A
210         LD   HL,TABLA
220         LD   B,0
230         LD   C,A
240         ADC  HL,BC
250         LD   B,(HL)
260         INC  HL
270         LD   D,(HL)
280         LD   E,0
320         JR   CLS3_1
330 TABLA   DEFB #40,#58
340         DEFB #48,#59
350         DEFB #50,#5A
390 CLS3_1  LD   H,B
400         LD   L,0
410 ...........
```

A partir de la línea 410 continúa como está arriba. Hemos intentado mantener la numeración en lo que fuera posible, pero al desaparecer líneas, quedan huecos vacíos entre los números. Esta rutina ocupa menos bytes que la original y, además, es más rápida. ¿Qué más se puede pedir?

Ahora sí, empecemos con los ejemplos específicos de este capítulo. El primero de ellos sirve para intercambiar entre sí dos de los tres bloques de pantalla (los que borraba la rutina anterior). Existen tres posibilidades: se puede intercambiar el primer bloque con el segundo, el segundo con el tercero o el primero con el tercero (nótese que intercambiar, por ejemplo, el segundo bloque con el primero es lo mismo que intercambiar el primero con el segundo, por tanto, en los tres casos expuestos están todas las posibilidades). Los bloques que se intercambiarán dependerán del contenido del acumulador que podrá ser «4», «5» ó «6». Se han elegido estos valores para que esta rutina sea compatible con la anterior; veamos un cuadro resumido:

```
A=4 => 1<->2
A=5 => 2<->3
A=6 => 1<->3
```

La idea es que, mezclando estas rutinas y otras que veremos, se pueda llegar a hacer un auténtico «procesador de pantalla» que haga un gran número de operaciones dependiendo del valor de «A».

Llamamos archivo de pantalla a los bytes cuyas direcciones están comprendidas entre 4000h (16384) y 57FFh (22527) ambos inclusive y que contienen los 49152 pixels que componen una imagen. Asimismo, llamamos archivo de atributos a los bytes con direcciones comprendidas entre 5800h (22528) y 5AFFh (23295) que contienen los atributos de color, brillo y parpadeo para cada grupo de 64 pixels.

Podemos dividir cada uno de estos archivos en tres zonas iguales; cada una de ellas contendrá 2048 bytes en el caso del fichero de pantalla y 256 bytes en el de atributos.

Intercambiar entre sí dos de estas zonas implica que los datos contenidos en cada una de sus direcciones de memoria se intercambian entre sí, y esto ocurre tanto para el archivo de pantalla como para el de atributos. Necesitaremos, por tanto, dos bucles; uno que intercambie los bytes del fichero de pantalla y otro que haga lo mismo con los bytes correspondientes del fichero de atributos. Para no andar manejando números engorrosos, asignaremos una etiqueta para la dirección de comienzo de cada una de estas zonas; las etiquetas serán «DIS_1», «DIS_2» y «DIS_3» para el archivo de pantalla y «ATT_1», «ATT_2» y «ATT_3» para el de atributos; sus direcciones (según se definen en el programa) serán:

```
660 DIS_1  EQU  #4000
670 DIS_2  EQU  #4800
680 DIS_3  EQU  #5000
690 ATT_1  EQU  #5800
700 ATT_2  EQU  #5900
710 ATT_3  EQU  #5A00
```


P/V; pone 1 - si BC-1 es diferente de cero
P/V; pone 0 - en cualquier otro caso

CICLOS DE MEMORIA:

Si «BC» diferente de 0 y «A» diferente del octeto
5

Si «BC» igual a 0 o «A» igual al octeto
4

CICLOS DE RELOJ:

Si «BC» diferente de 0 y «A» diferente del octeto
21

Si «BC» igual a 0 o «A» igual al octeto

EJEMPLO:

```
CPIR
```

Contenido del acumulador (octeto buscado):

(A):	10000010	82h

Contenido del par de registros «HL»:

(H):	11100000	E0
(L):	10000000	80h

Contenido del par de registros «BC»:

(B):	00000000	00h
(C):	00000111	07h

Contenido de las 7 posiciones de memoria a partir de E080h:

E080h	01111110	7Eh
	01111111	7Fh
	10000000	80h
	10000001	81h
	10000010	82h
	10000011	83h
	10000100	84h

Instrucción

CPIR:	11101101	EDh
	10110001	B1h

El contenido del registro «A» no ha variado con la ejecución.

Contenido del par de registros «HL» después de la ejecución:

(H):	11100000	E0h
(L):	10000100	84h

Contenido del par de registros «BC» después de la ejecución:

(B):	00000000	00h
(C):	00000011	03h

El contenido de las 7 posiciones de memoria a partir de E080h no ha variado después de la ejecución.

Indicadores de condición después de la ejecución:

S Z H P/V N C

0 1 × 0 × 1 1 ×

Se ha encontrado un octeto igual al contenido del acumulador en la posición de memoria E084h, por lo que la ejecución se ha detenido en este punto; «Z» está a «1» para indicar que se ha encontrado el octeto; «P/V» está a «1» porque «BC» no ha llegado a valer cero; «HL» contiene la dirección del octeto cuyo contenido es igual al del acumulador.

CPD

OBJETO:

Compara el contenido del registro acumulador con el octeto de memoria direccionado por el contenido par de registros «HL». Si la comparación, es verdadera se activará el indicador de condición «Z». A continuación decrementa en 1 los pares de registros «HL» y «BC».

Funciona igual que CPI pero realizando la búsqueda desde el final del bloque hacia el principio.

CODIGO DE MAQUINA:

11101101	EDh
10101001	A9h

INDICADORES DE CONDICION QUE AFECTA:

S; pone 1 - si el resultado es negativo
S; pone 0 - en cualquier otro caso

Z; pone 1 - si el registro «A» es igual al octeto
Z; 0 - en cualquier otro caso

H; pone 1 - si no hay acarreo desde el bit 3
H; pone 0 - en cualquier otro caso

N; pone 1 - siempre

P/V; pone 1 - si BC-1 es diferente de cero
P/V; pone 0 - en cualquier otro caso

CICLOS DE MEMORIA:

4

CICLOS DE RELOJ:

16

EJEMPLO:

```
CPD
```

Contenido del registro acumulador:

(A):	00100000	20h

Contenido del par de registros «HL»:

(H):	10111001	B9H
(L):	10001000	88h

Contenido del octeto de memoria B988h:

B988h:	00000010	02h

Contenido del par de registros «BC»:

(B):	00000000	00h
(C):	00000000	00h

Instrucción:

CPD:	11101101	EDh
	10101001	A9h

El contenido del registro «A» no ha variado con la ejecución.

Contenido del par de registros «HL» después de la ejecución:

(H):	10111001	B9h
(L):	10000111	87h

El contenido de la posición de memoria B988h no ha variado con la ejecución.

Contenido del par de registros «BC» después de la ejecución:

(B):	11111111	FFh
(C):	11111111	FFh

Indicadores de condición después de la ejecución:

S	Z		H		P/V	N	C
0	0	×	0	×	0	1	×

CPDR

OBJETO:

Compara el contenido del registro acumulador con el octeto de memoria direccionado por el contenido del par de registros «HL». Si la comparación es verdadera se activará el indicador de condición «Z». A continuación decrementa en 1 los pares de registros «HL» y «BC». Si el resultado de la comparación es falso y el contenido del par de registros «BC» no es cero se repite la instrucción.

La instrucción termina cuando el par de registros «BC» alcanza el valor cero o el resultado de la comparación es verdadero. Ver Figura 8-5.

Tenga en cuenta que si el par de registros «BC» es cero antes de la ejecución de la instrucción esta se repetirá para 64K octetos salvo que encuentre un resultado verdadero. Esto es debido a que primero decrementa y luego compara, al decrementar 1 a 0000h, en el par de registros «BC» quedaría el valor FFFFh.

Las interrupciones no paran la ejecución de esta instrucción por lo que se atenderían cuando terminase.

CODIGO DE MAQUINA:

11101101	EDh
10111001	B9h

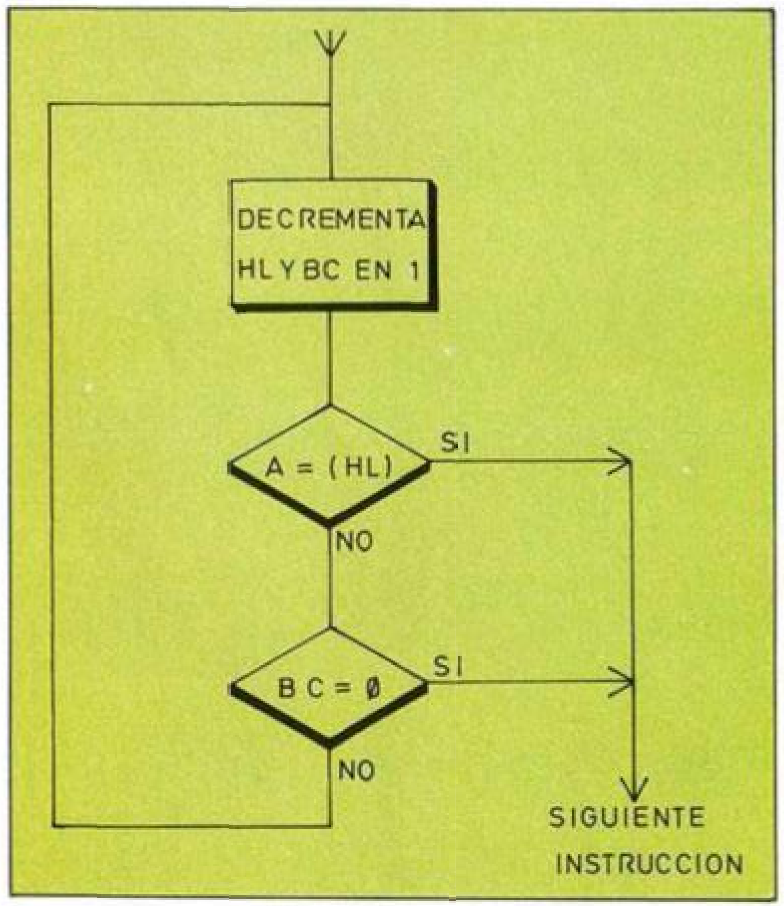


Fig. 8-5. Organigrama de la instrucción CPDR.

ciones fundamentales, como las de rotación o subrutinas; no obstante, estamos ya en disposición de escribir rutinas que puedan pasar a formar parte de la «biblioteca» particular de cada lector.

Programar en Assembler no es tarea fácil o, cuando menos, resulta bastante trabajoso, por lo que, los programadores, han ideado métodos para no tener que repetir el mismo trabajo varias veces. Uno de esos métodos es lo que se denomina: «biblioteca de rutinas». Programando en Assembler, hay una serie de tareas que se repiten en casi todos los programas; por ejemplo, borrar la pantalla, leer una tecla, hacer una pausa de «x» segundos, imprimir un dato en pantalla, buscar un dato en memoria, etc. La mayoría de los programadores escriben estas rutinas una sola vez y las guardan para utilizarlas en sucesivos programas; de esta forma, acaban teniendo una enorme colección de rutinas ya escritas y, cuando tienen que escribir un programa, combinan unas cuantas de ellas y ya tienen escrito más de la mitad del programa. A esta colección de rutinas se le denomina «biblioteca»; es conveniente que la mayoría de ellas estén escritas de forma reubicable, ya que así, resultará más fácil combinarlas.

Al hacer el curso, se nos ocurrió que los ejemplos pudieran servir para que cada lector iniciara, con ellos, su biblioteca particular. En este capítulo hemos preparado tres ejemplos que seguramente utilizará muchas veces. El primero de ellos sirve para intercambiar entre sí bloques de pantalla, y pretende ilustrar el manejo de las instrucciones «EX» y «EXX». El segundo permite guardar pantallas en zonas altas de memoria y recuperarlas desde allí; evidentemente, ilustra a la perfección el manejo de «LDIR». Por último, el tercer ejemplo es quizá el más útil, se trata de una rutina que le permitirá buscar cualquier conjunto desde 1 hasta 50 octetos consecutivos en una zona determinada de memoria, devolviendo la dirección inicial, si los encuentra, o «0» si no los encuentra (porque no existan, claro); como ya habrá supuesto, el núcleo principal de este ejemplo es la instrucción «CPIR».

Antes de pasar a ver el primero de estos ejemplos, hemos creído importante reproducir aquí una idea apuntada por el joven programador británico David Webb en su libro «Advanced Spectrum Machine Languaje». La idea es genial, principalmente, por su sencillez; se trata de una rutina que utiliza la instrucción «LDIR» para borrar la pantalla. Cuando borramos la pantalla, lo que hacemos es, precisamente, cargar un cero en cada una de sus direcciones (desde 4000h hasta 57FFh inclusive); la forma normal de hacerlo, sería con un bucle que fuera cargando «ceros» en estas direcciones una por una; así lo hicimos en nuestra rutina para borrar la pantalla por trozos (aún no habíamos visto la instrucción «LDIR»). David Webb propone otra forma más rápida de hacerlo: se trata de cargar la primera dirección en «HL», cargar un «0» en esta dirección, cargar la segunda dirección en «DE», cargar la longitud menos uno en «BC» y hacer «LDIR»; el primer byte será copiado en todos los restantes.

Este método puede ser utilizado cada vez que se quiera llenar una zona de memoria con un determinado octeto. Un ejercicio interesante para el lector sería modificar la rutina que borra la pantalla por trozos, de forma que lo haga utilizando «LDIR»; recuerde que cuando borre el fichero de atributos, no será un «0» lo que tendrá que cargar, sino el valor de los atributos permanentes en curso, que está almacenado en la variable del sistema «ATTR-P» (dirección 23693). Si modificamos las rutinas «BUC-1» y «BUC-2» de este ejemplo, quedarán así:

```
410         LD    BC,#07FF
420 BUC_1   PUSH  DE
430         LD    (HL),L
440         LD    D,H
450         LD    E,1
460         LDIR
470 BUC_2   POP   HL
480         LD    D,H
490         LD    E,1
500         LD    A,(23693)
510         LD    (HL),A
520         LD    C,#FF
530         LDIR
540         RET
```

En la línea 420 guardamos el contenido de «DE» que es la dirección de inicio de la zona de atributos; luego, en la línea 470, la recuperamos cargándola en «HL». Dado que el octeto menos significativo de todas las direcciones que usamos es «00», hemos supri-

empezaríamos a buscar el 00h, es de suponer que muchos números comiencen por 00h, por lo tanto, es más fácil organizar una búsqueda hacia atrás buscando el número más significativo, en este caso el 72h; una vez encontrado se analiza si el que le sigue hacia atrás es F3h , 74h, 00h y 00h.

Es norma habitual y aconsejable que los campos numéricos se justifiquen a la derecha y los literales a la izquierda, quiere esto decir que si en 5 octetos se quiere almacenar el literal «PEPE», se haga.

P	E	P	E	

y si se quiere almacenar el número 74F372h se haga,

00	00	74	F3	72

Una vez esto claro se podría aplicar como norma general a practicar que la búsqueda de una cadena literal se realice hacia delante y la búsqueda de una cadena numérica se realice hacia atrás. Ver organigrama de Figura 8-10.

El tipo de búsqueda visto hasta aquí es secuencial, es decir, se va mirando secuencialmente, bien sea hacia delante o hacia atrás unos octetos consecutivos de memoria.

Podría darse el caso de tener los datos organizados de forma que estén mezclados numéricos y literales además de diferentes tipos de información, eso sí, en una estructura previamente fijada. Por ejemplo, supongamos que se tiene información sobre libros y se coloca de la siguiente manera:

20 octetos - AUTOR
20 octetos - TITULO
15 octetos - ISBN
4 octetos - PRECIO
6 octetos - FECHA DE COMPRA
1 octeto - CODIGO TEMATICO
TOTAL 66 octetos

A esta estructura se la llama registro de datos, cada una de sus subdivisiones se llaman campos y una serie de registros forman un fichero, todo ello independientemente del soporte en el que estén (memoria, cinta o disco). Más adelante en el curso se hablará de forma de organizar datos, bases de datos, etc. pero de momento es bueno diferenciar campo, registro y fichero.

Continuemos con las instrucciones de búsqueda que es lo que nos ocupa. Supongamos que, en un fichero así organizado, queremos saber si existe y dónde está un libro llamado «MOMO».

Un método sería buscar en todo el fichero la cadena de caracteres «MOMO» pero este tiene dos problemas fundamentales:

1. Hay que rastrear todo el fichero conscientes de que en más de las dos terceras partes de él no existe la información referente al título.

2. Una vez encontrada la cadena de caracteres «MOMO« no estaríamos seguros de haber encontrado un autor o un título.

El método más válido sería el secuencial con índice, esto es, se buscaría sólo en los octetos del campo TITULO. Para lo cual una vez direccionado el primer registro se sumaría 20 a dicha dirección; se buscaría en los 20 octetos restantes la cadena de caracteres «MOMO» y si no se encuentra se sumaría a la dirección actual el tamaño del registro de datos menos 20, en este caso 46 con lo cual se estaría direccionando el campo TITULO del siguiente registro.

Observe que la palabra registro tiene dos usos muy utilizados en informática y muy diferentes en su función. Uno es el ya más familiar registro de CPU y otro es el registro de datos. No siempre que se emplea esta palabra se especifica si es CPU o datos, pero el contexto y el tema lo aclararán suficientemente.

Hasta aquí, hemos hecho una primera aproximación al manejo de tablas de datos; existen otros métodos más complejos que permiten, incluso, manejar tablas en las que los datos no tengan todos la misma longitud; pero no es el momento de abrumar al lector entrando excesivamente en profundidades; en un capítulo posterior, cuando veamos las técnicas de programación, haremos un estudio exhaustivo con ejemplos de lo que son y cómo se crean las bases de datos.

Ejemplos

El primer ejemplo que vimos en el curso, se limitaba a cargar el registro «BC» con un número y retornar; decíamos entonces que no parecía algo muy vistoso para lo que suelen evocar las palabras «código máquina»; a medida que hemos ido aprendiendo más instrucciones, nos ha sido posible aumentar el nivel de complejidad de nuestros ejemplos e ir haciendo cosas cada vez más útiles. Aún nos faltan por ver algunas instruc-

INDICADORES DE CONDICION QUE AFECTA:

S; pone 1 - si el resultado es negativo
S; pone 0 - en cualquier otro caso

Z; pone 1 - si el registro «A» es igual al octeto
Z; pone 0 - en cualquier otro caso

H; pone 1 - si no hay acarreo desde el bit 3
H; pone 0 - en cualquier otro caso

N; pone 1 - siempre

P/V; pone 1 - si BC-1 es diferente de cero
P/V; pone 0 - en cualquier otro caso

CICLOS DE MEMORIA:

Si «BC» diferente de 0 y «A» diferente del octeto:
5

Si «BC» igual a 0 o «A» igual al octeto:
4

CICLOS DE RELOJ:

Si «BC» diferente de 0 y «A» diferente del octeto:
21

Si «BC» igual a 0 ó «A» igual al octeto:
16

EJEMPLO:

CPDR

Contenido del acumulador:

(A): 11111111 FFh

Contenido del par de registros «HL»:

(H): 10000100 84h
(L): 00100011 23h

INSTRUCCIONES DE INTERCAMBIO

Código Fuente	Hexadecimal	Decimal
EX DE,HL	EB	235
EX AF,AF'	08	8
EX (SP),HL	E3	227
EX (SP),IX	DD,E3	221,227
EX (SP),IY	FD,E3	253,227
EXX	D9	217

Fig. 8-6. Tabla de codificación para las instrucciones de intercambio.

INSTRUCCIONES DE TRANSFERENCIA

Código Fuente	Hexadecimal	Decimal
LDD	ED,A8	237,168
LDDR	ED,B8	237,184
LDI	ED,A0	237,160
LDIR	ED,B0	237,176

Fig. 8-7. Tabla de codificación para las instrucciones de transferencia.

Contenido del par de registros «BC»:

(B): 00000000 00h
(C): 00000111 05h

Instrucción:

CPDR: 11101101 EDh
10111001 B9H

Contenido de las 5 posiciones de memoria anteriores a 8423h:

00110011 33h
00110100 34h
00110101 35h
00110110 36h
8423h: 00110111 37h

El contenido del registro «A» no ha variado con la ejecución.

Contenido del par de registros «HL» después de la ejecución:

(H): 11100000 E0h
(L): 00011110 1Eh

Contenido del par de registros «BC» después de la ejecución:

(B):	00000000	00h
(C):	00000000	00h

El contenido de las 5 posiciones de memoria anteriores a E080h no han variado después de la ejecución.

Indicadores de condición después de la ejecución:

S Z H P/V N C

0 0 × 0 × 0 1 ×

En este caso, no se ha encontrado ningún octeto cuyo contenido sea igual al del acumulador, por tanto, la instrucción ha terminado cuando «BC» ha llegado a valer cero. El indicador «Z» está a «0» porque no se ha encontrado el octeto y el «P/V» está, también, a «0» porque el registro «BC» ha llegado a valer cero.

Tablas de codificación

A continuación, vamos a ver las tablas que nos indican el código máquina que corresponde a cada instrucción; recuerde que éstas son las tablas que utilizaremos para ensamblar las rutinas «a mano».

En la Figura 8-6 tenemos la tabla correspondiente a las instrucciones de intercambio; en la Figura 8-7, la correspondiente a las de transferencia y, finalmente, en la Figura 8-8 tenemos la tabla correspondiente a las instrucciones de búsqueda.

Hemos añadido otra tabla en la Figura 8-9 que constituye un resumen de la forma en que estas instrucciones afectan a los indicadores, así como el número de bytes que ocupan y el número de ciclos de memoria y reloj que emplean.

INSTRUCCIONES DE BUSQUEDA

Código Fuente	Hexadecimal	Decimal
CPD	ED,A9	237,169
CPDR	ED,B9	237,185
CPI	ED,A1	237,161
CPIR	ED,B1	237,177

Fig. 8-8. Tabla de codificación para las instrucciones de búsqueda.

GRUPO DE INTERCAMBIO, TRANSFERENCIA Y BUSQUEDA

NEMONICO	INDICADORES S	Z	x	H	x	P/V	N	C	No. DE BYTES	CICLOS MEM.	REL.
EX DE,HL	.	.	.	.	.	.	.	.	1	1	4
EX AF,AF'	.	.	.	.	.	.	.	.	1	1	4
EX (SP),HL	.	.	.	.	.	.	.	.	1	5	19
EX (SP),IX	.	.	.	.	.	.	.	.	2	6	23
EX (SP),IY	.	.	.	.	.	.	.	.	2	6	23
EXX	.	.	.	.	.	.	.	.	1	1	4
LDI	.	.	x	0	x	↕	0	.	2	4	16
LDIR	.	.	x	0	x	0	0	.	2	5(4)	21(16)
LDD	.	.	x	0	x	↕	0	.	2	4	16
LDDR	.	.	x	0	x	0	0	.	2	5(4)	21(16)
CPI	↕	↕	x	↕	x	↕	1	.	2	4	16
CPIR	↕	↕	x	↕	x	↕	1	.	2	5(4)	21(16)
CPD	↕	↕	x	↕	x	↕	1	.	2	4	16
CPDR	↕	↕	x	↕	x	↕	1	.	2	5(4)	21(16)

NOTAS:

1.- Los signos tienen el siguiente significado:

"↕": El indicador cambia de valor de acuerdo con el resultado de la instrucción.

"x": El bit adquiere un estado indeterminado.

".": El indicador no es afectado por la instrucción, por lo que conserva su anterior contenido.

"0": El indicador se pone siempre a cero.

"1": El indicador se pone siempre a uno.

Fig. 8-9. Tabla de indicadores y ciclos para las instrucciones de intercambio, transferencia y búsqueda.

Métodos de búsqueda

Las instrucciones de búsqueda se usan normalmente para localizar en parte o en la totalidad de la memoria un valor que se supone que existe. También se suelen usar, aunque menos, para asegurarse de que un determinado valor no existe.

Cuando el valor que se quiere buscar ocupa un octeto, la aplicación de las instrucción es inmediata, y una vez ejecutada sabremos si dicho valor existe o no, en caso de que exista también sabremos dónde se encuentra.

El problema se complica cuando se quiere buscar un valor que ocupa más de un octeto. Por ejemplo en una tabla donde existen nombres propios, queremos buscar «PEPE». La forma de actuar en un caso como éste sería:

a) Buscar primero una «P».

b) Si no se encuentra no hay ningún «PEPE».

c) Si se encuentra habrá que comparar si los siguientes caracteres son «E», «P» y «E», lo cual se puede hacer con la instrucción CP o bien con las de búsqueda que incrementan el registro que se usa como índice.

Si lo que se busca es un valor numérico que ocupa más de un octeto es de mayor utilidad emplear otro método, a saber: supongamos que se quiere buscar el número 00074F372h en una tabla que contiene números de 5 octetos como máximo. Si se emplea el método anterior

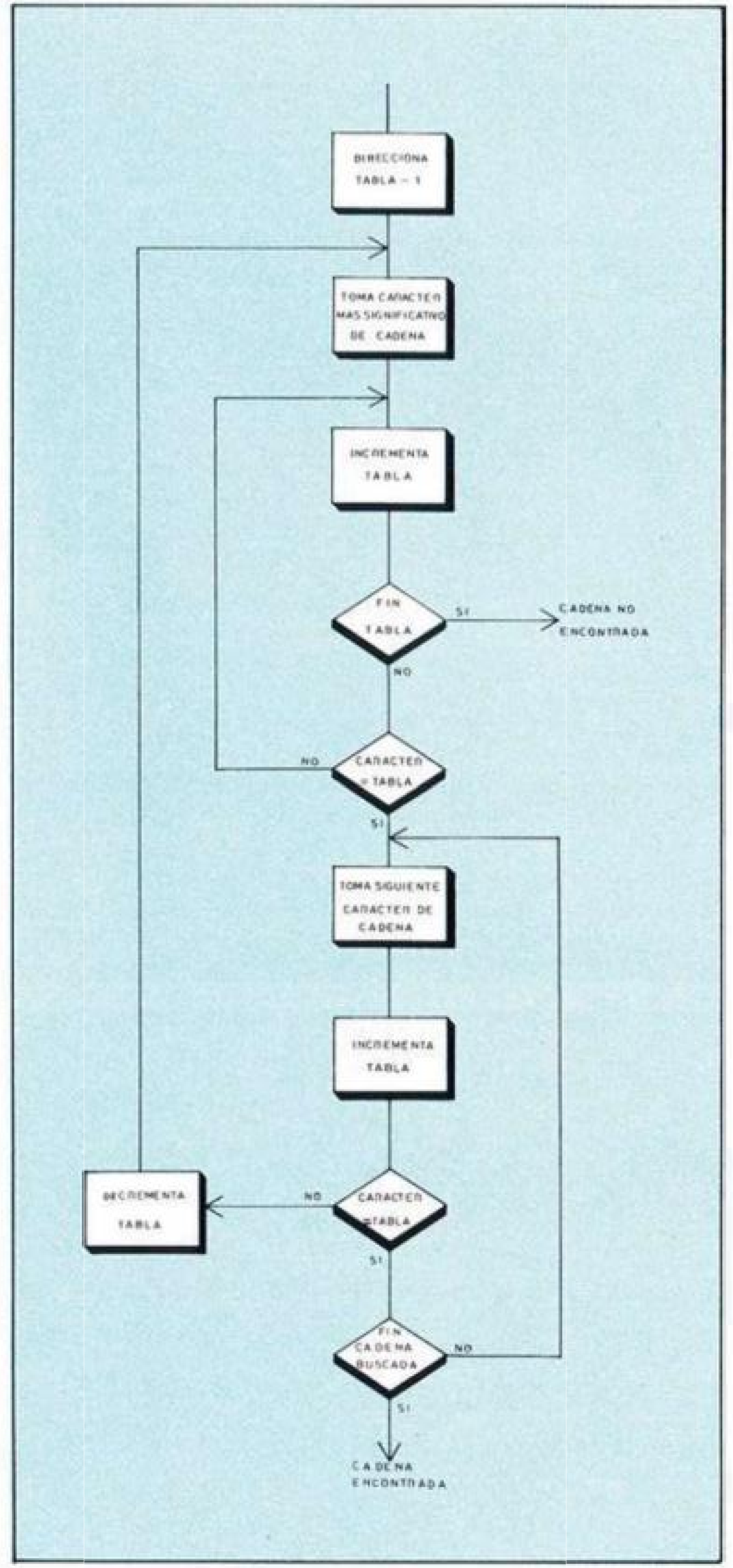


Fig. 8-10. Organigrama de una búsqueda en cadena.

& Juegos ESTRATEGIA

¡Juegos de Simulación Estratégica para poner a prueba su inteligencia y vivir la emoción de situaciones reales!

¿Se atreve Ud. a dirigir la Campaña del Desierto y derrotar a Rommel antes de que lo hiciera el General Montgomery en el Alamein?

Un WAR GAME de estrategia que reúne todos los condimentos necesarios para hacer de él un auténtico *«plato fuerte»*, inteligencia, emoción, sorpresa y realismo.

Sólo 1.125 pts. (I.V.A. incluido)

RATAS del DESIERTO

Operación Norte de Africa

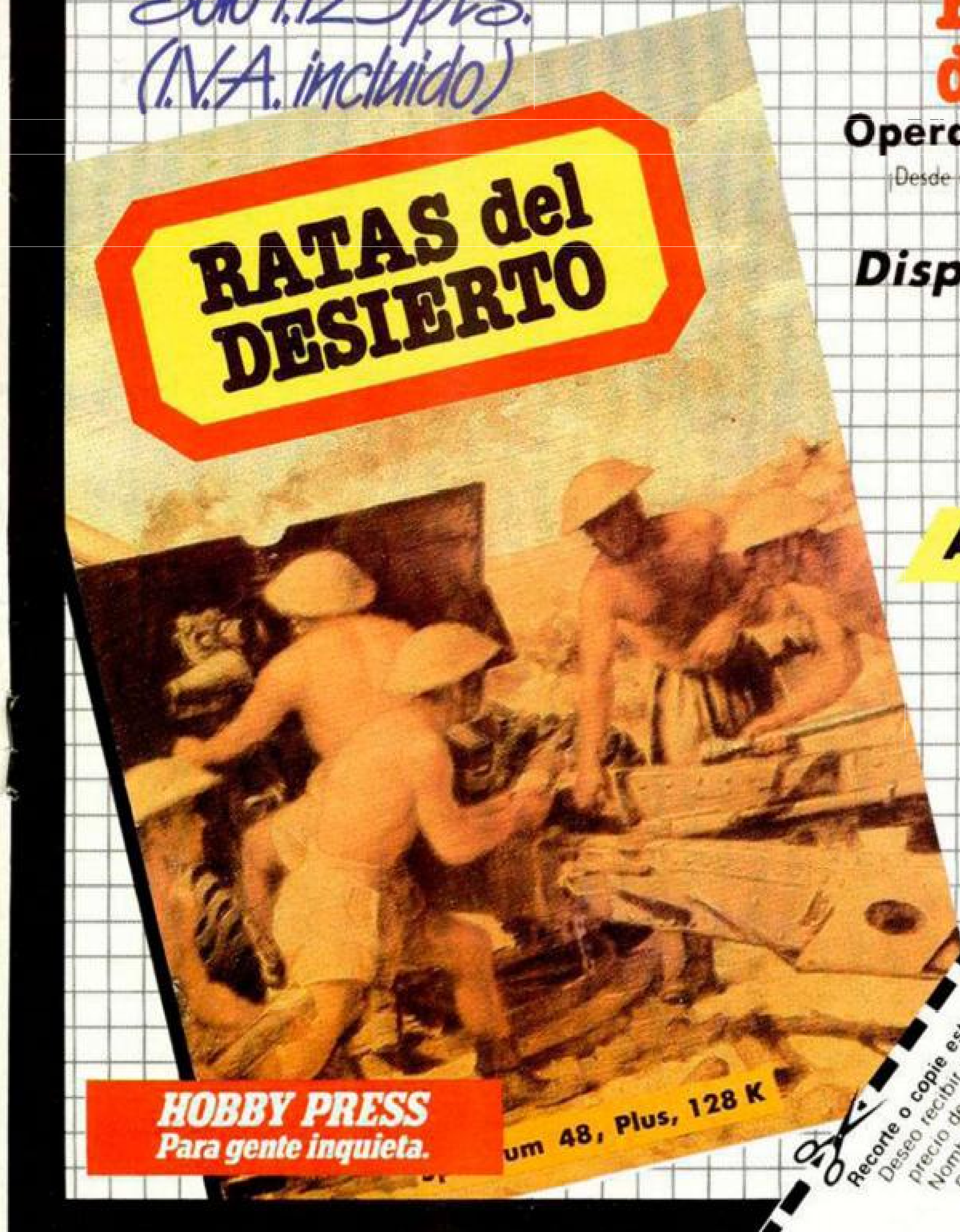
¡Desde uno a tres jugadores!

Disponible para

Spectrum

y

Amstrad



HOBBY PRESS
Para gente inquieta.

Recorte o copie este cupón y envíelo a Hobby Press. S. A. Apdo. de Correos 54.062 Madrid.
Deseo recibir en mi domicilio, **sin gastos de envío alguno por mi parte,** la cinta **RATAS DEL DESIERTO** al precio de 1.125 ptas. (IVA incluido)
Nombre
Dirección
Localidad — Provincia
Código — Teléfono — Edad
El modelo que elijo es para:
Amstrad — Spectrum
La forma de pago elegida es la que señalo con una cruz
Giro postal n.º — Talón nominativo a Hobby Press, S. A. — Tarjeta Visa n.º
Fecha de caducidad de la tarjeta — Fecha y firma

LA ZONA DE VARIABLES (y III)

Rafael PRADES

Concluimos esta serie sobre la Zona de Variables con un programa de utilidad en la depuración y conocimiento de las variables y las direcciones donde están almacenadas, así como algunos ejemplos de utilización práctica.

El programa número 1, cuyo listado se publicó por razones de espacio en el número anterior, es un complemento de todo lo explicado sobre las variables, a parte de su utilidad en la depuración de programas, ya que permite conocer todas las variables definidas, así como las direcciones a partir de la cual están almacenadas.

Para ejecutarlo utilizar la sentencia:

```
GO TO 9000
```

Las matrices van seguidas de los paréntesis ( ), que permiten una rápida y sencilla localización, y las variables de control de bucle del símbolo #.

```
NOMBRE          DIRECCION
vars            > 25335
paso            > 25344
n#              > 25353
z               > 25372
pepe            > 25378
a$()            > 25387
c()             > 25795
v#              > 25901
salvar          > 25920
x               > 25931
z()             > 25937
tron            > 26003
>>>>>>>>>>>>>>> FIN <<<<<<<<<<<<<<<
```

Ejemplo de utilización del Programa 1.

Aplicación

Una vez conocidos los códigos con los que se almacenan las variables y la forma en que está estructurada dicha zona, podemos resolver un problema que se plantea al grabar, con la sentencia «SAVE», una cadena de caracteres como matriz numérica; es decir:

```
LET a$="juanito"
SAVE "nombre" DATA a$( )
```

El S.O. permite que se realice el almacenamiento de esta variable como matriz, aunque a$ no esté dimensionada como tal.

Si inicializamos el ordenador mediante un reset, NEW, RANDOMIZE USR Ø o una nueva conexión, podremos cargar de nuevo los datos almacenados, asignándolos a la misma variable o a otra distinta, con el comando:

```
LOAD "nombre" DATA a$ ( )
```

Pero cuál será nuestra sorpresa cuando no podamos ni visualizar, ni fragmentar, ni asignar un nuevo valor a la variable, ya que el S.O. nos muestra el siguiente mensaje de error.

```
3 Subscript wrong
```

Analizando detenidamente la memoria con el programa que nos visualiza los contenidos de cada dirección, observamos que ha sido almacenada de la siguiente manera:

193	−96 = 97 = "a"
7	longitud
O	7 bytes
1O6	= "j"
117	= "u"
97	= "a"
11O	= "n"
1O5	= "i"
116	= "t"
111	= "o"

Los tres bits más significativos del primer byte forman el código de una matriz alfanumérica (11Ø en binario) y los restantes bytes tienen la estructura de una cadena de caracteres. ¿Cómo solucionar esta incompatibilidad entre código y estructura? La forma más sencilla es la de cambiar el código del primer byte para adaptarlo a cadena de caracteres, ya que únicamente se diferencian en el bit de mayor peso (11ØXXXXX-Ø1ØXXXXX).

Para realizar este cambio, podemos utilizar dos métodos; el primero consiste en realizarlo mediante comandos directos, es decir, localizar con ayuda del programa «1», publicado en el número anterior, la dirección de comienzo de la variable y luego ejecutar

```
POKE dirección, PEEK
dirección-128
```

El segundo método, consiste en cambiarlo con ayuda de un pequeño programa en código máquina que se almacena en el buffer de impresión. El programa número «2» se encarga de almacenar el código objeto en dicha área. Puede utilizarlo en cualquiera de sus programas, teniendo en cuenta que su ejecución se realiza con la sentencia:

```
RANDOMIZE USR 23296
```

La explicación de este programa es sencilla:

En el par de registros «HL» se carga la dirección de comienzo de la zona de variables, almacenadas en la dirección «23627» decimal o «5C48» hexadecimal (variable del sistema «VARS»).

En el acumulador (registro «A») se

PROGRAMA 2

```
10 REM CARGADOR A$
20 FOR n=23296 TO 23312
30 READ dato
40 POKE n,dato
50 NEXT n
60 DATA 42,75,92,126,254,193,4
0,6,205,184,25,235,24,245,54,65,
201
```

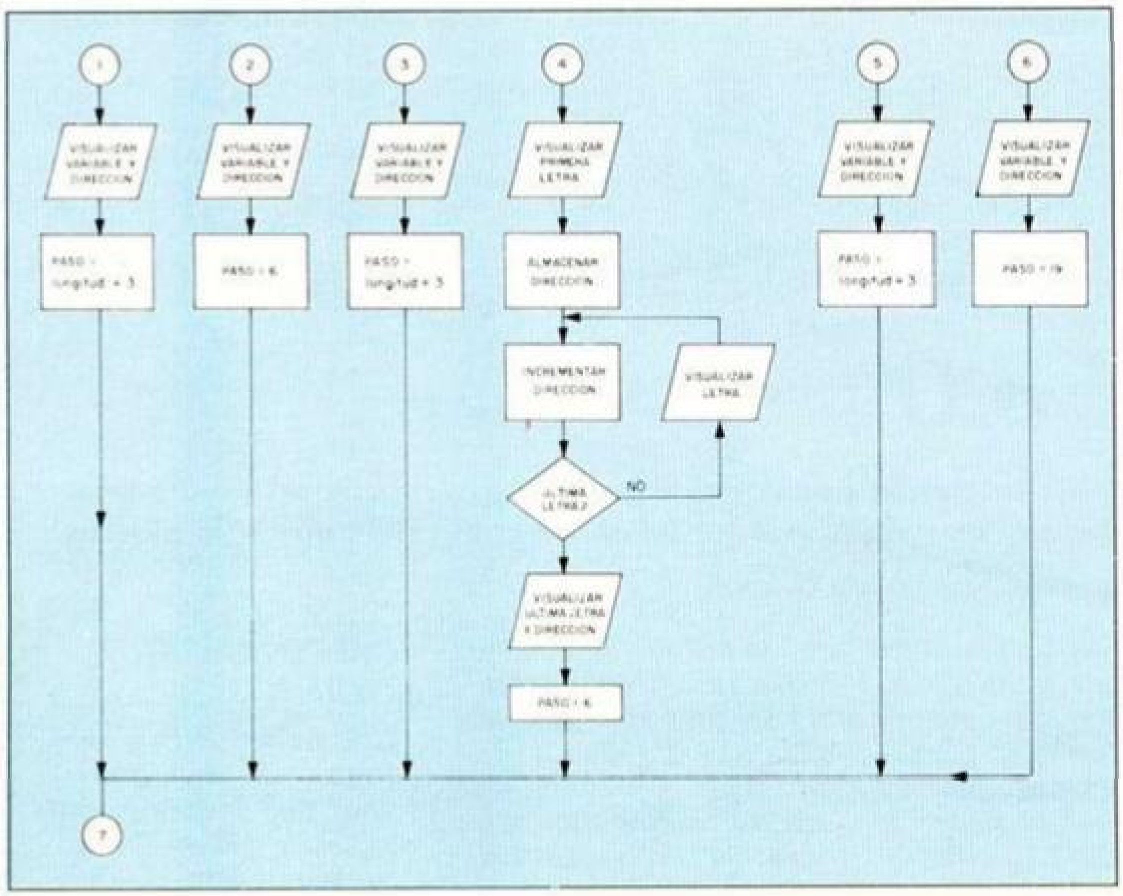


Fig. 3. Pograma 1. Visualización de variables y direcciones.

```
DIRECCION COD. OBJETO      CODIGO FUENTE
========= =========== =====================

  5B00       2A4B5C           LD   HL,(#5C4B)
  5B03       7E         START LD   A,(HL)
  5B04       FEC1             CP   #C1
  5B06       2806             JR   Z,FIN
  5B08       CDB819           CALL #19B8
  5B0B       EB               EX   DE,HL
  5B0C       18F5             JR   START
  5B0E       3641       FIN   LD   (HL),#41
  5B10       C9               RET
===========================================
```

Listado Assembler del programa "2" con direcciones y datos en hexadecimal.

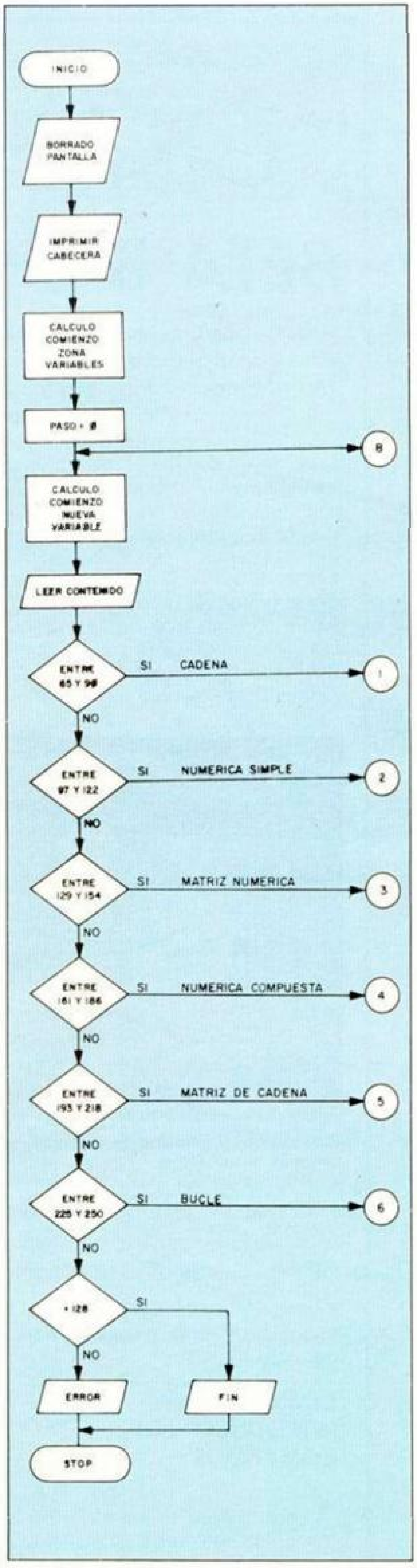
carga el contenido de la dirección almacenada en «HL».

Posteriormente se compara con el valor de la variable a buscar, si es a$ será «193» decimal o «C1» hexa. Si coinciden, se almacena en la dirección apuntada por «HL» el valor correspondiente al código de cadena de caracteres («65» en decimal o «41» en hexa. para la variable a$).

En caso contrario, llama a la rutina de la ROM «NEXT-ONE», ubicada en la dirección «5048» dec. o «19B8» hexa. Básicamente esta rutina calcula el comienzo de una nueva variable, si en «HL» hay una dirección perteneciente a esta zona. El resultado lo devuelve en el par de registros «DE», por tanto será necesario efectuar un intercambio con «HL», para volver a cargar en el acumulador el código de la siguiente variable y efectuar una nueva comparación.

Al final, si la variable es encontrada, el programa retorna al BASIC.

Si en lugar de a$ se utiliza otra variable, será necesario modificar el programa para alterar los valores de comparación y sustitución, localizados en los datos 6(«193») y 16(«65») de la líneas de DATAS.



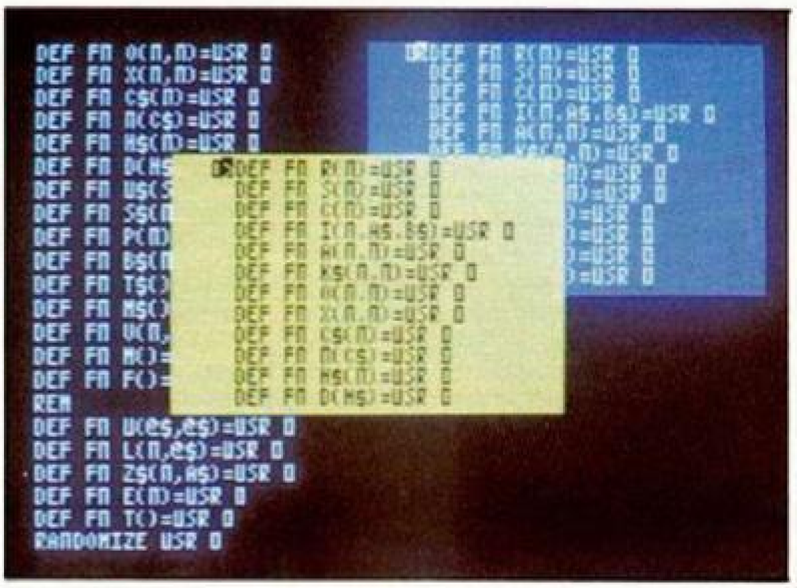
Pograma 1. Bucle de lectura.

BETA BASIC 3.0

El Beta-Basic es un «toolkit» que se caracteriza por varios aspectos: gran facilidad de manejo, total transparencia del sistema hacia el usuario e incorporación de comandos que sólo existen en ordenadores más grandes.

Una vez cargado el programa, devuelve inmediatamente el control al usuario presentando el cursor tal y como si nada hubiese pasado; sin embargo, a partir de ahora disponemos de un ordenador con una mayor potencia en lenguaje Basic.

El manual de instrucciones que acompaña al programa es claro, explicando detalladamente cada comando o función con ejemplos de manejo, lo que es un punto a favor de este «toolkit».



Podremos usar infinidad de ventanas formateadas del modo que más nos interese.

COMO SE INTRODUCEN LOS COMANDOS

La primera sorpresa ..e a la hora de introducir los diversos comandos que se han añadido al Basic,, sólo hay que pulsar el modo gráfico y una tecla: el comando se presenta en pantalla, sin necesidad de complicadas llamadas a rutinas en CM. Esta facilidad hace que su utilización sea completamente transparente; es como si tuviéramos otro modo extendido, además del que ya posee el Spectrum.

Ya habrás pensado sin duda que con este software no puedes utilizar los GDU... pues no, hay uno de los comandos que determina esto: KEYWORDS, éste puede estar en cinco modos distintos. Así, con KEYWORDS Ø se acceden a los GDU, con KEYWORDS 1 a los nuevos comandos. Los números desde el 2 al 4 indican la forma de introducir con la sola pulsación de una tecla o letra a letra como en otros ordenadores. Esto también es útil a la hora de introducir comandos definidos al igual que en lenguaje LOGO o FORTH.

El cursor que normalmente está en modo «K» lo podemos cambiar fácilmente al modo «L» pulsando «Space», esto es muy útil a la hora de introducir los comandos definidos; también podemos hacer la operación inversa, esto es, cambiar el cursor de «L» a «K» con la combinación de teclas «Simbol» + «Enter».

Otro aspecto destacable es que la forma en que el listado se presenta en pantalla es definible, pudiéndolo hacer más profesional y claro; hay para elegir un total de 6 formas distintas.

También es definible la resolución de cada carácter en la pantalla en pixels!!!, esto nos sirve para formatear la pantalla en caracteres de 4x7, por ejemplo, con lo que tendremos 64 columnas por 25 líneas. Por último, y para colmo, con respecto al tema de presentación en pantalla, podemos definir las dimensiones de ésta a nuestro gusto, reservando una parte para listados y otra para ejecutar el programa igual que el QL.

ALMECENAMIENTO EXTERNO

Los usuarios que dispongan de microdrive están de enhorabuena con este programa ya que, por medio de otro comando se puede definir por defecto el medio de almacenamiento externo que se va a usar, bien cassette, microdrive, RS232, o red local. Una vez definido sólo hay que usar los comandos consabidos de cassette y el sistema se encargará de mandar la información al lugar adecuado, evitándose la engorrosa síntaxis del Interface 1.

UTILIDADES DEL BASIC

Un paquete tan potente se quedaría cojo si no tuviera comandos para edición de listados Basic, comandos que existen en ordenadores mayores, tales como AUTO, RENUM, DELETE (para borrar líneas de Basic), REF (para buscar literales en un listado), JOIN (para unir líneas distintas), KEYIN (para CREAR líneas del Basic desde dentro del mismo programa), EDIT. A partir de ahora dispondremos de todas estas utilidades.

Posee también la facilidad TRACE para depurar programas y el ON ERROR que habrás visto en otros ordenadores.

El manejo de variables

es muy amplio también, pudiéndolas editar con el comando EDIT de idéntica forma a como lo hace el 128K, asignarles un valor por defecto, sacar un listado de todas las que existen actualmente, etc...

SALTOS Y BUCLES

Ahora no tendremos que limitarnos al uso de los bucles FOR-NEXT, sino que dispondremos de un amplio repertorio: DO, LOOP, EXIT IF, ELSE, WHILE, UNTIL, ON, etc. En el manual se explican claramente estos comandos, utilizables hasta ahora sólo en ordenadores grandes.

PROCEDIMIENTOS

De la misma forma en que se definen nuevos comandos, se pueden definir procedimientos que se pueden llamar desde nuestros programas, y también se le pueden definir parámetros de entrada y salida, incluso se permiten los procedimientos recursivos, muy codiciados en programación de inteligencia artificial.

Ya comentamos que existe lo que podríamos llamar un segundo modo extendido con el modo gráfico, pues bien, también hay un tercer modo extendido representado esta vez con el cursor con un asterisco, este modo se consigue pulsando a la vez «Simbol shift» y «Space», y sirve para introducir los macro comandos que hayamos definido en las teclas.

COMANDOS GRAFICOS

Tampoco se queda corto en lo que respecta a comandos gráficos, posee instrucciones de scroll de ventanas, un fill super-inteligente, por supuesto también es capaz de imprimir en alta resolución pixel a pixel, otra función es ALTER para manejar los atributos de pantalla de una forma más racional. Todo esto puede hacer que al programador de juegos le de un síncope.



En este ejemplo se ve como hay una ventana para listados y el resto de la pantalla para datos.

OTRAS POSIBILIDADES

Otros comandos no se pueden clasificar dentro de los apartados anteriores. CLOCK, por ejemplo, permite poner en la pantalla un reloj en todo momento, programar una alarma para que nos avise a cierta hora o hacer que un programa en Basic se pare en cierto momento y bifurque a otra rutina.

Una particularidad muy importante es que el manejo del cursor se hace más racional pudiéndolo mover dentro del listado hacia arriba o abajo rápidamente, produciéndose un scroll hacia abajo del listado cuando el cursor llega a la parte superior de la pantalla. Una vez hayamos editado una línea, podemos mover el cursor por ésta más fácilmente que de costumbre.

La tecla «Break» se ha hecho más potente pudiendo parar incluso programas en CM.

LAS FUNCIONES

Además de comandos, disponemos de nuevas funciones, las cuales están definidas en la línea O del listado Basic. Esta línea no se puede borrar a no ser que no vayamos a usar ninguna de las funciones definidas.

Entre éstas se puede encontrar un SENO y un COSENO más rápido y preciso que el estandar. Una función para convertir números de hexadecimal a decimal y de decimal a binario, otra para ver el número de puntos que ha rellenado el comando FILL, etc..., hasta un total de 26.

Resumiendo: bajo el nombre de Beta-Basic 3.0 se puede conseguir un paquete de utilidades por un precio más que aceptable que amplia el Basic de nuestro ordenador con 66 comandos nuevos y 26 funciones, lo cual no está nada mal, y que puede cumplir las exigencias del más estricto programador en lenguaje Basic.

La forma más rápida de hacerse con este paquete de utilidades es ponerse en contacto con BETASOFT, 92 Oxford Road, Moseley, BIRMINGHAM. Londres.

CALCULADORA POR INTERRUPCIONES

Bernardo MORCEGO

Esta revista ya ha publicado algunos programas cuyo único objetivo era facilitar el cálculo de expresiones matemáticas. Sin embargo, ninguno de estos programas era capaz de trabajar simultáneamente con otro en Basic, cosa que es muy útil para culquier programador despistado que no recuerde cuánto son 8 x 12...

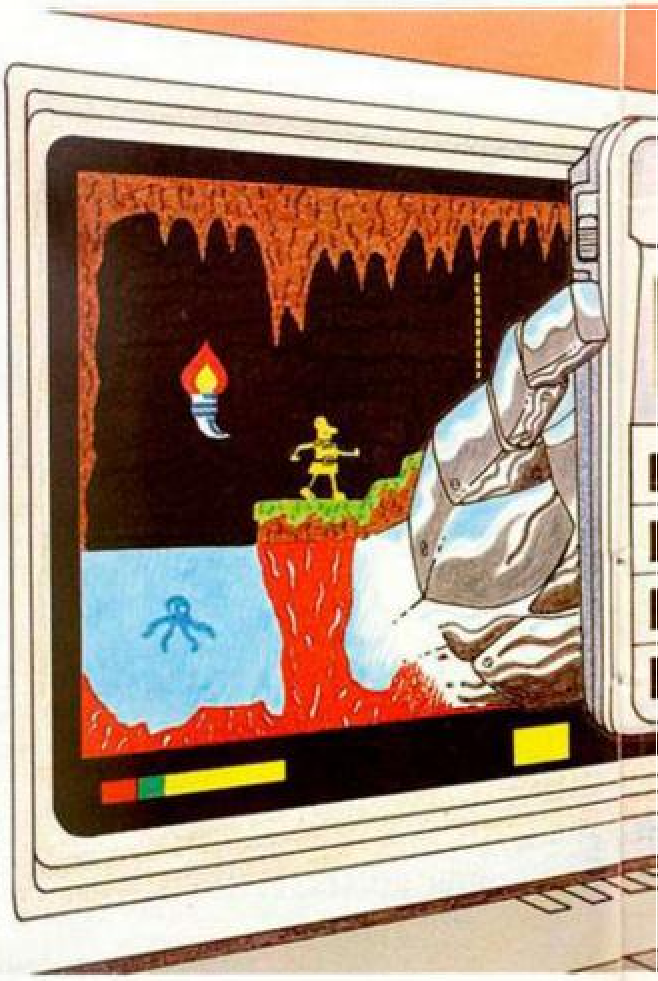
El programa listado a continuación ofrece esta posibilidad gracias a la utilización del código máquina y a la de las interrupciones del Spectrum. Para evitar que se extendiera demasiado, la calculadora sólo es capaz de multiplicar, dividir, sumar, restar y hacer la raíz cuadrada de cualquier número que no tenga más de 15 dígitos, sin embargo, esto lo puede hacer del mismo modo que lo haría una calculadora comercial. Es decir, con la posibilidad de operar resultados y corregir números entrados incorrectamente.

Cuando llamamos al programa, sólo un cierto número de teclas serán admitidas. Estas teclas son las siguientes:

- Números: del Ø al 9 y el punto decimal (Symbol Shift + M).
- Operadores:
  - − (Symbol Shift + J)
  - + (Symbol Shift + K)
  - * (Symbol Shift + B)
  - / (Symbol Shift + V)
  - ↑ (Symbol Shift + H)

  Enter (da el resultado de una operación)
- Funciones: STOP (Symbol Shift + A vuelve al Basic)

  NOT (Symbol Shif + S borra de la pantalla el número tecleado previamente, si lo hubiera).



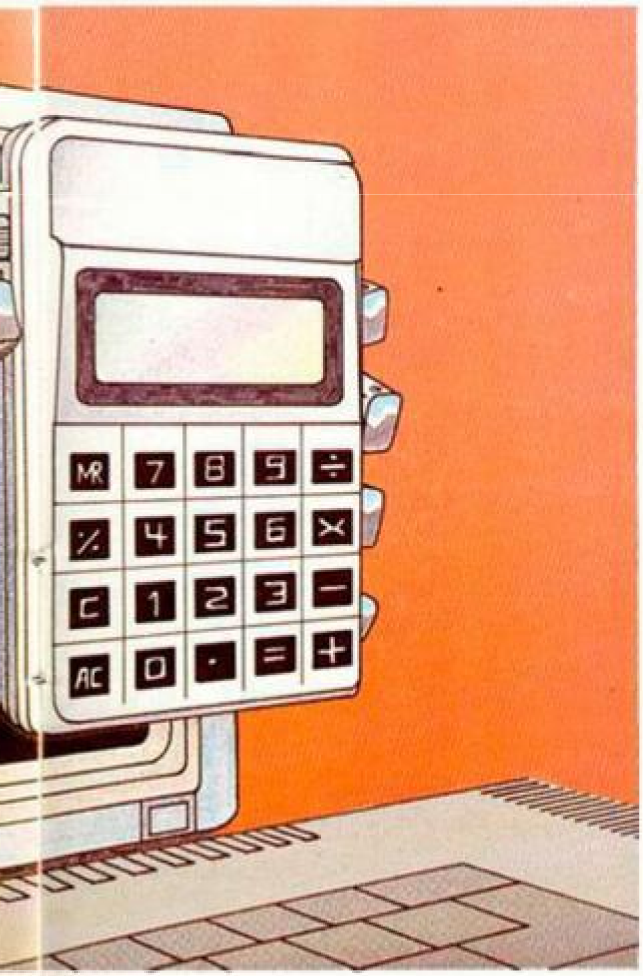
F. L. FRONTAN

Utilización del programa

Para utilizar el programa no hay más que introducir, valiéndonos del Cargador Universal de Código Máquina, el listado 1, haciendo un DUMP en la di-

DESENSAMBLE DEL PROGRAMA «CALCULADORA». LISTADO 3

```
10 ; PROGRAMA CALCULADORA
20 DRAW   EQU  #24BA
30 FLAG   EQU  64254
40 CHADD  EQU  23645
50 BUFF   EQU  64228
60 SPOSN  EQU  23688
70 DF_CC  EQU  23684
80 LASTK  EQU  23560
90 LAST1  EQU  64252
100 FLAG2 EQU  64253
110 VARIS EQU  64000
120 DECFP EQU  #2C88
130 CHA+1 EQU  #0074
140 CLEFT EQU  #0A23
150       ORG  64000
160 INTAC LD   A,250
170       LD   I,A
180       IM   2
190       RET
200 INTDE IM   1
210       RET
220       ORG  64255
230       DEFW COMZO
240 COMZO PUSH AF
250       PUSH HL
260       PUSH BC
270       PUSH DE
280       EXX
290       PUSH AF
300       PUSH HL
310       PUSH BC
320       PUSH DE
330       EXX
340       CALL #02BF
350       LD   HL,23611
360       BIT  5,(HL)
370       JR   Z,EXIT
380       LD   A,(LASTK)
390       SUB  95
400       CP   1
410       JR   C,START
420 EXIT  EXX
430       POP  DE
440       POP  BC
450       POP  HL
460       POP  AF
470       EXX
480       POP  DE
490       POP  BC
500       POP  HL
510       POP  AF
520       RST  #38
530       RETI
540 ;**CUERPO DEL PROGRAMA**
550 START CALL INTDE
560       EXX
570       PUSH HL
580       EXX
590       LD   A,110
600       LD   (23677),A
610       LD   A,169
620       LD   (23678),A
630       LD   BC,#007B
640       LD   DE,#0001
650       CALL DRAW
660       LD   BC,#1300
670       LD   DE,#FF00
680       CALL DRAW
690       LD   BC,#007B
700       LD   DE,#00FF
710       CALL DRAW
720       LD   BC,#1300
730       LD   DE,#0100
740       CALL DRAW
750       EXX
760       POP  HL
770       EXX
780       LD   HL,23552
790       LD   DE,VARIS
800       LD   BC,145
810       LDIR
820       LD   HL,23612
830       RES  0,(HL)
840       CALL CHVARS
850       CALL CL
860       XOR  A
870       LD   (FLAG),A
880       LD   (FLAG2),A
890       LD   (BUFF),A
900       CALL #2C88
910       CALL VARS
920       CALL CL
930 TECLAS CALL #02BF
940       LD   HL,23611
950       BIT  5,(HL)
960       JR   Z,TECLAS
970       LD   A,(LASTK)
980       LD   C,A
990       SUB  #2E
1000      CP   1
1010      JR   C,NUMS1
1020      SUB  #02
1030      CP   10
1040      JR   C,NUMS
1050      LD   A,C
1060      SUB  #2A
1070      CP   2
1080      JR   C,OPER1
1090      SUB  3
1100      CP   1
1110      JR   C,OPER2
1120      SUB  2
1130      CP   1
1140      JR   C,OPER3
1150      SUB  47
1160      CP   1
1170      JR   C,OPER4
1180      LD   A,C
1190      SUB  #0D
1200      CP   1
1210      CALL C,ENTER
1220      SUB  #B6
1230      CP   1
1240      JP   C,NOT
1250      SUB  #1F
1260      CP   1
1270      JP   C,STOP
1280      JR   TECLAS
1290 NUMS1 LD  HL,BUFF
1300      LD   B,15
1310 LOOP LD   A,(HL)
1320      SUB  13
1330      CP   1
1340      JR   C,NUMS
1350      SUB  #21
1360      CP   1
1370      JR   C,TECLAS
1380      INC  HL
1390      DJNZ LOOP
1400 NUMS LD   A,(DF_CC)
1410      SUB  78
1420      CP   15
1430      CALL CLS1
1440      JR   NC,TECLAS
1450      LD   HL,FLAG
1460      BIT  6,(HL)
1470      PUSH HL
1480      CALL NZ,#2BF1
1490      POP  HL
1500      RES  6,(HL)
1510      CALL PRINT
1520      JR   TECLAS
1530 OPER1 LD  HL,FLAG2
1540      INC  A
1550      LD   (HL),A
1560      JR   OPERS
1570 OPER2 LD  HL,FLAG2
1580      SET  2,(HL)
1590      JR   OPERS
1600 OPER3 LD  HL,FLAG2
1610      SET  3,(HL)
1620      JR   OPERS
1630 OPER4 LD  HL,FLAG
1640      SET  4,(HL)
1650      DEC  HL
1660 OPERS INC HL
1670      RES  6,(HL)
1680      CALL ENTER
1690      LD   HL,FLAG2
1700      LD   B,(HL)
1710      LD   DE,FLAG
1720      LD   A,(DE)
1730      AND  128
1740      OR   B
1750      OR   128
1760      LD   (DE),A
1770      XOR  A
1780      LD   (HL),A
1790      XOR  A
1800      DEC  HL
```



rección 4ØØØØ. Una vez hecho esto, provocaremos un BREAK (CAPS SHIFT+SPACE) en el Cargador y luego RUN. Esto hará que empiece de nuevo desde la línea 1, sin perder el contenido de la memoria anteriormente volcado. Ahora introducimos el listado 2 y hacemos un DUMP en la dirección 4Ø255. Por último, salvar todos los bytes juntos con la opción SAVE OBJETO, indicando 4ØØØØ como dirección y 981 como número de bytes.

A la hora de cargarlo para su utilización, es imprescindible hacer CLEAR 63999:LOAD""CODE 64ØØØ, los afortunados poseedores de un Ensamblador, pueden utilizar el listado 3.

No es reubicable, por lo que sólo se podrá utilizar tomando como dirección inicial la 64ØØØ.

Para activar el programa basta con teclear RANDOMIZE USR 64ØØØ y, a partir de entonces, hasta que se produzca un NEW, cada vez que pulsemos "_" (Symbol Shift + Ø) aparecerá en la esquina superior derecha un recuadro que representa la pantalla de nuestra calculadora.

Desde aquel momento las únicas teclas que aceptará el ordenador serán las anteriormente señaladas. También existe una forma menos radical que el NEW para desactivar el programa, y es tecleando RANDOMIZE USR 64ØØ7.

No es necesario teclear RANDOMIZE USR 64ØØØ cada vez que queramos utilizar la calculadora, sino sólo una vez, y a partir de entonces Symbol Shift + Ø será suficiente para poder hacer nuestros cálculos.

Aprovechamiento de la ROM

La ROM es un extenso programa, creado por Sinclair Research, en el que se pueden encontrar muchísimas rutinas útiles a la hora de hacer nuestros programas. Por ejemplo: la ROM incluye una rutina que se encarga de todo tipo de cálculos, ya sean numéricos, de cadenas alfanuméricas o lógicos. En el programa calculadora se hace uso de muchas de estas rutinas con un fin principal: el ahorro de memoria. He aquí una relación de las 12 rutinas a las que se llama repetidamente a lo largo del programa:

1. DRAW (24BAh): Dibuja una línea entre dos puntos dados.
2. DEC-TO-FP (2CB8h): Transforma un número decimal a su formato en coma flotante.
3. KEYBOARD (Ø2BFh): Lee el teclado y devuelve en el registro A y en LAST-K el código de la tecla presionada.
4. CH-ADD+1 (ØØ74h): Incrementa el contenido de la variable del sistema CH-ADD.
5. PRINT-OUT (Ø9F4h): PRINTa en pantalla el carácter correspondiente al código almacenado en LAST-K.
6. BEEPER (Ø3B5h): Produce

```
1810        LD   (HL),A
1820        CALL PRINTY
1830        JP   TECLAS
1840 ENTER  LD   A,(FLAG2)
1850        AND  31
1860        JR   Z,ENT2
1870 ENT1   LD   A,(BUFF)
1880        CP   14
1890        RET  C
1900        CALL VARS
1910        CALL #2C88
1920        LD   A,(FLAG)
1930        AND  31
1940        RET  Z
1950        JR   ENT3
1960 ENT2   LD   A,(FLAG)
1970        AND  31
1980        RET  Z
1990        JR   ENT1
2000 ENT3   LD   HL,FLAG
2010        SET  7,(HL)
2020        CALL CALC
2030        CALL CHVARS
2040        LD   A,32
2050        LD   (LASTK),A
2060        CALL PRINT1
2070        CALL VARS
2080        LD   A,(FLAG)
2090        OR   192
2100        AND  192
2110        LD   (FLAG),A
2120        CALL KEY1
2130        RET
2140 NOT    CALL VARS
2150        CALL CL
2160        CALL BEP
2170        JP   TECLAS
2180 STOP   LD   HL,VARIS
2190        LD   DE,23552
2200        LD   BC,145
2210        LDIR
2220        CALL BEEP
2230        CALL KEY1
2240        LD   A,32
2250        LD   (LASTK),A
2260        CALL INTAC
2270        JP   EXIT
2280 ;**SUBRUTINAS**
2290 PRINT  LD   HL,(CHADD)
2300        LD   A,(LASTK)
2310        LD   (HL),A
2320        CALL CHA+1
2330        LD   A,(LASTK)
2340 PRINT1 PUSH AF
2350        CALL #09F4
2360        POP  AF
2370 BEP    LD   HL,23609
2380        LD   D,0
2390        LD   E,(HL)
2400        LD   HL,#00CB
2410        CALL #0385
2420        CALL KEY1
2430        RET
2440 VARS   LD   HL,BUFF
2450        LD   (CHADD),HL
2460        LD   HL,5650
2470        LD   (SPOSN),HL
2480        LD   HL,16462
2490        LD   (DF_CC),HL
2500        RET
2510 CLS1   LD   HL,FLAG
2520        BIT  7,(HL)
2530        RET  Z
2540 CL     LD   HL,LASTK
2550        LD   DE,LAST1
2560        LD   A,(HL)
2570        LD   (DE),A
2580        LD   A,32
2590        LD   (HL),A
2600        LD   B,15
2610 LOPCLS PUSH BC
2620        PUSH AF
2630        CALL #09F4
2640        POP  AF
2650        POP  BC
2660        DJNZ LOPCLS
2670        CALL VARS
2680        LD   HL,BUFF
2690        LD   A,13
2700        LD   B,15
2710 LPCLS  LD   (HL),A
2720        INC  HL
2730        DJNZ LPCLS
2740        LD   HL,FLAG
2750        RES  7,(HL)
2760        LD   DE,LASTK
2770        LD   HL,LAST1
2780        LD   A,(HL)
2790        LD   (DE),A
2800        RET
2810 BEEP   LD   DE,#0030
2820        LD   HL,#0666
2830        CALL #0385
2840        RET
2850 KEY1   CALL #028E
2860        LD   A,E
2870        INC  A
2880        CP   1
2890        JR   NC,KEY1
2900        LD   HL,23611
2910        RES  5,(HL)
2920        RET
2930 PRINTY LD   A,(FLAG)
2940        AND  15
2950        RET  Z
2960        PUSH AF
2970        CALL CHVARS
2980        POP  AF
2990        CP   2
3000        JR   NC,NEX
3010        LD   A,120
3020        LD   (LASTK),A
3030        CALL PRINT1
3040        CALL VARS
3050        RET
3060 NEX    CP   5
3070        JR   NC,NEXT
3080        ADD  A,41
3090        CALL PRINT1
3100        CALL VARS
3110        RET
3120 NEXT   LD   A,45
3130        CALL PRINT1
3140        LD   BC,#A174
3150        CALL #22E5
3160        LD   BC,#A574
3170        CALL #22E5
3180        JP   VARS
3190 CHVARS CALL VARS
3200        LD   A,(23689)
3210        INC  A
3220        LD   (23689),A
3230        LD   A,(23684)
3240        SUB  32
3250        LD   (23684),A
3260        RET
3270 CALC   LD   A,(FLAG)
3280        AND  31
3290        CP   2
3300        JR   NC,SIG
3310        RST  #28
3320        DEFB #04
3330        DEFB #38
3340        JR   CAL1
3350 SIG    CP   3
3360        JR   NC,SIG1
3370        RST  #28
3380        DEFB #0F
3390        DEFB #38
3400        JR   CAL1
3410 SIG1   CP   5
3420        JR   NC,SIG2
3430        RST  #28
3440        DEFB #03
3450        DEFB #38
3460        JR   CAL1
3470 SIG2   CP   9
3480        JR   NC,SIG3
3490        RST  #28
3500        DEFB #05
3510        DEFB #38
3520        JR   CAL1
3530 SIG3   RST  #28
3540        DEFB #28
3550        DEFB #38
3560 CAL1   CALL VARS
3570        CALL CLS1
3580        CALL #2BF1
3590        PUSH AF
3600        PUSH BC
3610        PUSH DE
3620        INC  HL
3630        INC  HL
3640        INC  HL
3650        INC  HL
3660        INC  HL
3670        LD   (23653),HL
3680        CALL #2DE3
3690        POP. DE
3700        POP  BC
3710        POP  AF
3720        CALL #2ABA
3730        LD   HL,FLAG
3740        SET  6,(HL)
3750 CAL3   RET
```

una nota según el valor de los registros HL y DE.

7. KEY-SCAN (Ø28Eh): Forma parte de la rutina KEYBOARD.

8. PLOT-SUB (22E5h): PLOTa un punto según el valor del registro BC.

9. FP-CALC (ØØ28h): Calculadora de números en formato de coma flotante.

10. STK-FETCH (2BF1h): Extrae un número del Calculador Stack y lo distribuye entre el juego principal de registros.

11. STK-STORE (2AB6h): Hace exactamente lo contrario que la anterior.

12. PRINT-FP (2DE3h): Escribe en pantalla el último valor del Calculator Stack.

En el hipotético caso de que la ROM no existiera y la calculadora tuviera que incluir todas estas rutinas, en vez de ocupar poco menos de 750 bytes ocuparía bastante más de 3000.

Funcionamiento del programa

El programa está dividido en dos partes principales: la primera forma el cuerpo y la segunda el conjunto de subrutinas. He aquí una lista de las subdivisiones de ambas partes, de sus funciones y de sus rasgos más característicos:

INTAC e INTDE activan y desactivan las interrupciones.

COMZO y EXIT son las rutinas a las que se accede cincuenta veces por segundo, encargadas de leer el teclado y entrar en el programa si la tecla presionada es Symbol Shift+Ø. En el caso de que algún usuario desee cambiar la tecla de acceso a la calculadora tendrá que hacer: POKE 64281 (código en ASCII de la tecla).

START es la entrada a la calculadora. Se encarga de poner al día las variables, dibujar la pantalla y borrar su contenido.

TECLAS lee el teclado y salta a las correspondientes direcciones según la tecla presionada.

NUMS1 y NUMS aceptan números (y la coma) y los escriben en pantalla.

OPER1, OPER2, OPER3, OPER4 y OPER5 tratan con los ordenadores. Su misión es poner al día la variable FLAG del programa y llamar a ENTER en caso de que sea necesario, además de llamar a la subrutina PRINTY.

ENTER, ENT1, ENT2 y ENT3 se encargan de llamar a la calculadora (subrutina CALC) en el caso de que dos números hayan sido entrados.

NOT borra la pantalla (llamando a CL) e inicializa las variables exceptuando FLAG y FLAG2.

STOP vuelve al BASIC restaurando todas las variables del sistema.

La siguiente lista pertenece a las subrutinas, que, al igual que en este programa, suelen estar ubicadas después del cuerpo del mismo.

PRINT Y PRINT1 se encargan de escribir en pantalla, de acuerdo con las variables que le indican el lugar, cualquier carácter dado, haciendo uso de PRINT-OUT de la ROM. Además produce el típico «click» de presión de una tecla y llama a KEY1.

VARS es la subrutina de inicialización de variables, en concreto CH-ADD, SPOSN y DF-CC son las afectadas.

CLS1 y CL borran 15 caracteres a partir de una posición dada y vacían el buffer auxiliar.

BEEP efectúa el sonido de retorno al BASIC.

KEY1 espera hasta que no haya ninguna tecla presionada.

PRINTY llama a CHVARS y escribe el signo matemático correspondiente a la tecla presionada.

CHVARS cambia las variables inicializadas en VARS para poder utilizar la fila superior del display de la calculadora.

CALC se encarga de calcular la relación numérica entre los operandos. Para ello utiliza la variable FLAG, que contiene, entre otros datos, el operador tecleado. También escribe el resultado en pantalla (llamando a PRINT-FP de la rom) y finalmente vuelve a la rutina que le ha llamado, que es únicamente ENTER.

LISTADO 1

```
 1   3EFAED47ED5EC9ED56C9  1676
```

LISTADO 2

```
 1   01FBF5E5C5D5D9F5E5C5  2024
 2   D5D9CDBF02213B5CCB6E  1325
 3   28093A085CD65FFE0138  827
 4   0DD9D1C1E1F1D9D1C1E1  1942
 5   F1FFED4DCD07FAD9E5D9  1935
 6   3E6E327D5C3EA9327E5C  938
 7   0178001101000CDBA2401  570
 8   00131100FFCDBA240178  842
 9   0011FF00CDBA24010013  719
10   110001CDBA24D9E1D921  1137
11   005C1150FA019100EDB0  998
12   213C5CCB86CD73FDCDEC  1536
13   FCAF32FEFA32FDFA32E4  1812
14   FACD882CCDD3FCCDECFC  2044
15   CDBF0221385CCB6E28F6  1181
16   3A085C4FD62EFE013837  863
17   D602FE0A384679D62AFE  1237
18   02385CD603FE01385DD6  985
19   02FE01385ED62FFE0138  979
20   5F79D600FE01DC42FCD6  1450
21   B6FE01DA89FCD61FFE01  1544
22   DA95FC18B521E4FA060F  1356
23   7ED600FE013809D621FE  1174
24   0138A32310F03A845CD6  1007
25   4EFE0FCDE6FC309421FE  1517
26   FAC876E5C4F128E1C886  1690
27   CD81FC188321FDFA3C77  1504
28   181421FDFACBD6180D21  1067
29   FDFACBDE180621FEFACB  1698
30   E62B23CB86CD42FC21FD  1502
31   FA4611FEFA1AE68080F6  1647
32   8012AF77AF2877CD37FD  1290
33   C338F83AFDFAE61F2814  1467
34   3AE4FAFE0ED8CDD3FCCD  1693
35   B82C3AFEFAE61FC81808  1283
36   3AFEFAE61FC810E421FE  1562
37   FACBFECD86FDCD73FD3E  1934
38   2032085CCDBEFCCDD3FC  1497
39   3AFEFAF6C0E6C032FEFA  1976
40   CD28FDC9CDD3FCCDECFC  2060
41   CDC3FCC388FB2150FA11  1617
42   005C019100EDB0CD1EFD  1139
43   CD28FD3E2032085CCD00  947
44   FAC31EFB2A5D5C3A085C  1111
45   77CD75003A085CFE5CDF4  1293
46   09F121395C16005E21C8  781
47   00CDB503CD28FDC921E4  1349
48   FA225D5C2112162208SC  804
49   214E4022845CC921FEFA  1171
50   CB7EC821085C11FCFA7E  1307
51   123E2077060FC5F5CDF4  1143
52   09F1C110F7CDD3FC21E4  1635
53   FA3E0D060F772310FC21  801
54   FEFACB8E11085C21FCFA  1549
55   7E12C911300021660BCD  756
56   B503C9CD8E027B3CFE01  1172
57   30F721385CCB0AEC93AFE  1369
58   FAE60FC8F5CD73FDF1FE  2008
59   02300C3E7832085CCDBE  789
60   FCCDD3FCC9FE053009C6  1635
61   29CDBEFCCDD3FCC93E20  1664
62   CDBEFC0174A1CDE52201  1394
63   74A5CDE522C3D3FCCDD3  1823
64   FC3A895C3C32895C3A84  1068
65   5CD6203284SCC93AFEFA  1375
66   E61FFE023005EF043818  893
67   1EFE033005EF0F381815  695
68   FE053005EF0338180CFE  900
69   093005EF0538180 3EF28  668
70   38CDD3FCCDE6FCCDF128  1900
71   F5C5D5232323232323265  965
72   5CCDE32DD1C1F1CD862A  1641
73   21FEFACBF6C900000000  1187
```

Posibles errores

Muchos cálculos producen un resultado «imposible», al menos en nuestro mundo de números reales, que son los únicos con los que trabaja la calculadora, y eso causa un retorno al BASIC desde la misma ROM. Por esta razón el retorno es forzado, pues ninguna de las variables del sistema que el programa utiliza son restauradas a sus valores iniciales. De esta forma, los efectos de un retorno forzado pueden resultar, tanto fatales como inocuos aunque siempre son no aptos para cardíacos. Evitar este tipo de retorno se reduca a dos sencillas reglas:

1.ª No hacer divisiones en las que el denominador sea Ø.

2.ª Evitar las sumas, restas o multiplicaciones en las que el resultado pueda ser mayor o menor que 1E+39 y −1E+39, respectivamente.

Cuando, a pesar de las precauciones del programador, sobreviene una situación de retorno forzado, la forma de volver al BASIC suele ser haciendo BREAK y posteriormente DELETE. Además, para poder volver a acceder a la calculadora es necesario reactivar el programa con un RANDOMIZE USR 64ØØØ. Sin embargo, es posible que el control del ordenador vuelva por sí solo o que, sencillamente, no vuelva.

Finalmente, resta por aclarar que la utilización del programa es factible no sólo desde el sistema BASIC, sino que también lo es desde cualquier otro programa, esté en código máquina o no, que no deshabilite las interrupciones y que no ocupe la zona donde se ubica la calculadora.

MICRO Manía
Sólo para adictos

VARIOS

Ultimamente os hemos venido ofreciendo los POKES y trucos de aquellos programas que podríamos denominar superjuegos: West Bank, Basketball, Herbert, Fairlight, Matchpoint, etc.

Pero no sólo de superjuegos vive el hombre, y estamos seguros de que tendréis por ahí cientos de juegos de esos, más o menos de segunda fila que ya habéis olvidado, o que hace algún tiempo fueron vuestro principal ingreso de diversión.

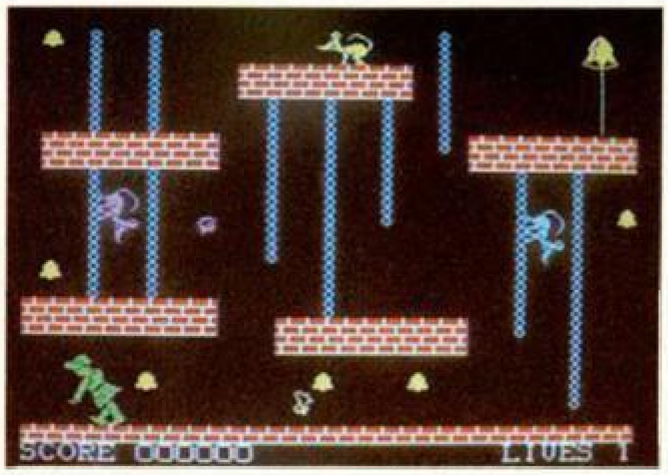
Así pues, ahora tenéis la oportunidad de no ser desagradecidos con ellos, y con los POKES de vidas infinitas que os vamos a ofrecer podréis sacarlos del fondo del cajón y volver a disfrutar un rato con ellos.

Ahí van, y esperamos que los tengáis todos:

GROUND ATTACK: POKE 29Ø63,Ø
KOSMIC KANGA: POKE 36212,Ø
LUNAR JETMAN: POKE 36965,Ø
PYRAMID: POKE 44685,Ø (Energía infinita)
MR. WINPY: POKE 33693,Ø
ZZOOM: POKE 24743,Ø
POKE 32692,Ø
TUTANKHAMUN: POKE 27783,Ø
ZIPZAP: POKE 54Ø65,Ø (Energía infinita)
ORION: POKE 37319,2Ø1
HORACIO SKI: POKE 2927Ø,Ø (Quita tráfico)







LA FAMILIA WALLY EN EL PARAISO
o los héroes también lloran (III)

Por Dessas Trosso

Nada, no hay por qué preocuparse. Como verás, la flecha ha dado en el blanco (en el negro, para ser más exacto), pero por el momento, aún no es posible rescatar a Wilma.

Ahora, tienes que ir al pozo de los deseos, subirte en él y quedarte encima de la cuerda. Utilízala para bajar hasta el fondo, donde encontrarás una botella. Hazte con ella y escala hacia el exterior por el muro derecho del pozo.



Coge el sacacorchos de la herrería y vuelve a hacerle una visita al cocodrilo. Usando el bolso que antes habías dejado preparado, pasa al otro lado la botella y el sacacorchos.

Sitúate enfrente del coco que hay en la cola del cocodrilo (parece un trabalenguas pero no lo es), y suelta el bolso de Wilma. Usa la botella y el sacacorchos para llenar ésta de aceite de coco. Coge el bolso de Wilma nuevamente y pasa a través de la boca del cocodrilo. Vete a la jungla y coge el hacha que allí encontrarás. En el sitio donde está la cabaña y el coche con ruedas de piedra, podrás afilar el hacha, usando para ello una de las ruedas del vehículo. Con el hacha a punto de caramelo podrás cortar la liana que mantiene atada a Wilma.

Vamos Wally, no te enrrolles, deja las efusiones para más tarde. ¡Recuerda que



Herbert aún está en la caldera!

Busca la lata que hay al lado de la rana gigante. Coge los Flip-flops que se encuentran en la oficina de correos.

Regresa al geiser (OLDE FAITHFULL) y repite el numerito de la liana, lo que te permitirá llenar la lata de agua caliente.

Ahora vas a la playa donde está el cangrejo.

¿Pero qué ocurre? El suelo cede bajo tus pies...

¡Te hundes, Wally! ¡Arenas movedizas!...

¿Conseguirá nuestro héroe escapar a tan horrible final...?

¿Estará la sopa de Herbert en su punto...?

No dejéis de leer el próximo capítulo de este lacrimógeno serial.



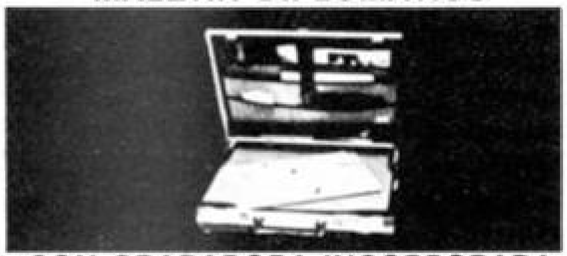
«LOS PROFESIONALES»





Mando automático para grabar las conversaciones telefónicas. Al descolgar, el magnetófono se pone en marcha. Al colgar, se para. Calidad Hi-Fi. Extensa gama. 1 hasta 8 líneas.

MALETIN DIPLOMATICO

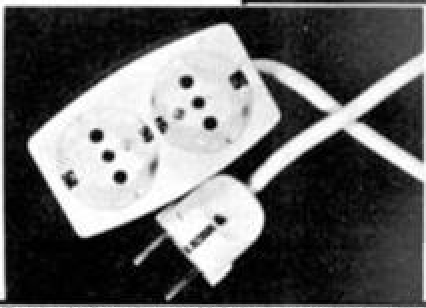


CON GRABADORA INCORPORADA TOTALMENTE AUTOMATICA.

Tamaño: 45 × 33 × 10, piel lujo.
Color: negro, marrón, rojo.
Cerraduras con números secretos.

Capta en un radio de 10 metros y graba automáticamente, abierto o cerrado, las conversaciones. No olvidará nada de las entrevistas importantes, técnicas, confidencias, etc.

IMPORTANTE: en 8 horas aseguramos el montaje del sistema completo en cualquier tipo de maletín, conservando todo el espacio de origen para los documentos.



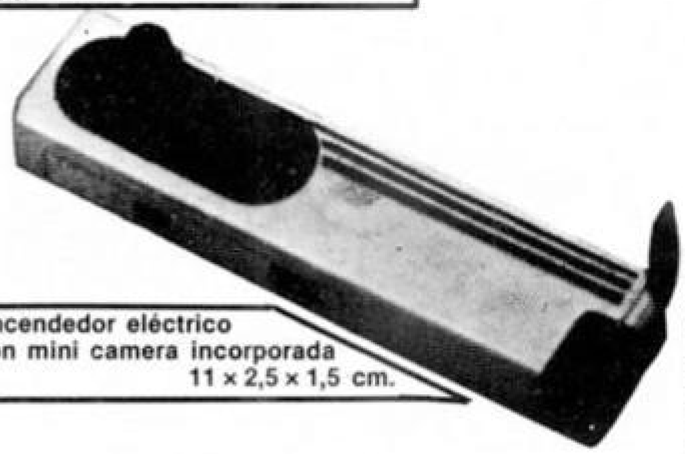
Enchufe hembra múltiple con emisora FM incorporada. Sin alimentación ni antena. Elimina totalmente el alterna. Super sensible.



Emisora telefónica subminiatura
14 × 14 × 7 mm. autonomía ilimitada. Alcance 300-500 m.

Binocular-Telefoto Camera
7 × 20 — Tele de 100 mm.
F 5,6 - 11 - 16 - 22
1/125 - 1/250 sec -
Tipo 110 - 400 ASA.

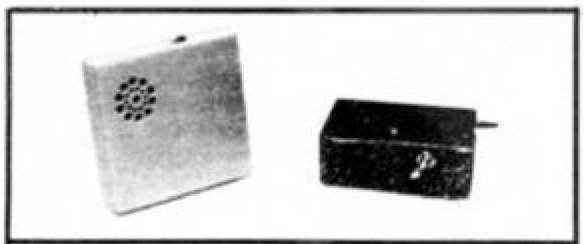




Encendedor eléctrico con mini camera incorporada
11 × 2,5 × 1,5 cm.

TOP SECRET

Todo nuestro material tiene una garantía de 6 o 12 meses.



SURVEYOR sistema revolucionario de escucha a distancia por medio del teléfono. Económico y muy práctico.



Para estar siempre localizable con el trasladador de llamadas.

TODA LA GAMA DE MATERIAL ELECTRONICO DE DETECCION, LOCALIZACION, NEUTRALIZACION, MICROEMISION, RECEPCION Y GRABACION AUTOMATICA, MODIFICADOR DE VOZ, VIDEO, LASSER, RADIOTELEFONO DE COCHE etc.

INFORMACION

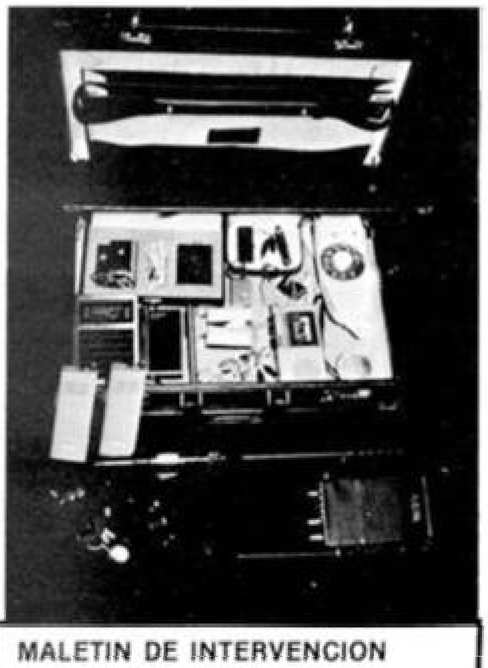
Deseo recibir el Catálogo y tarifa de los aparatos «TOP SECRET», por lo cual adjunto **500 Ptas.** en sellos en concepto de gastos.

Nombre
Apellidos
Calle .... Nº
Código Postal
Ciudad
Provincia

«QUICK, S.A.»
TOP ELECTRONIC
Diputación, 240, 6º, 7ª
08007 BARCELONA



Micro direccional + estetoscopio electrónico, para captar las conversaciones hasta 300 m, u oír a través de las paredes.



MALETIN DE INTERVENCION

Equipado con el material necesario para todo tipo de intervención.

CONSULTORIO

Mezclar imágenes

Quisiera saber si es posible, y de que manera podría hacerlo yo, un fundido del ordenador con imágenes de vídeo, por ejemplo, poner títulos a vídeo-clpis.

Tengo noticias de que alguien ha conseguido destruir un «QL» por software, ¿es cierto?

José A. TARRAGO - Sevilla

☐ La mezcla de imágenes sólo es posible si se dispone de un mezclador de video; esto se debe a que, a diferencia del audio, las dos imágenes a mezclar deben estar sincronizadas; amén de que el proceso de mezcla es considerablemente más complejo.

No hemos oído nada respecto a lo que usted se refiere sobre el «QL». El Sistema operativo de este ordenador se encuentra en ROM, por lo que es imposible destruirlo mediante un programa tipo «virus». Por otro lado, ¿cabe la posibilidad de que se pueda dañar permanentemente al ordenador con un software determinado? Bien, esta posibilidad es muy remota; en un ordenador bien diseñado, estas cosas no deben ocurrir; en el Spectrum, desde luego, no es posible.

Programas en Código Máquina

¿Cómo puedo saber la longitud de un programa en C/M, una vez metido en memoria, si sé su dirección?

¿Y si no sé su dirección?

Querría saber, también, algún «POKE» para saber cuanta memoria me queda al estar programando en C/M.

Carlos ROMEU - Madrid

☐ Suponemos que se refiere a un programa que ha cargado desde cassette, ya que si lo hubiera introducido por teclado, necesitaría saber su longitud antes de meterlo. Bien, el caso es que no se puede. Una vez que el programa está en memoria, es muy posible que haya echado a «correr» haciendo todas las «diabluras» posibles y, por supuesto, quitándonos el control. Para saber la longitud, tendrá que hacerlo antes de cargarlo, leyendo la cabecera de la cinta con un «lector de cabeceras» como el que publicábamos en el número 2, página 10 de MICROHOBBY; este lector no sólo le dará la longitud, sino también la dirección de inicio.

Si está programando en código máquina, no necesita que el ordenador le diga cuánta memoria le queda, lo sabrá usted si cuenta cuantos bytes de código ha «POKEado». Por otro lado, si está programando en Assembler, sabrá la memoria que le ocupa su programa cuando lo ensamble; no obstante, lo normal es no escribir el programa de un tirón, sino irlo haciendo a base de rutinas cortas que se ensamblarán y probarán por separado (tenga en cuenta que programar en Assembler es bastante diferente a hacerlo en Basic), con lo que no tendrá más que sumar lo que ocupa cada rutina.

Interface Centronics «INDESCOMP»

Tengo el interface Centronics de Indescomp para impresora, y quisiera saber si este periférico es compatible con el Interface 1 de Sinclair y los Microdrives, ya que tengo entendido que pueden surgir problemas durante el proceso de inicialización del ordenador.

Juan VANRELL - Madrid

☐ De momento, y para tranquilizarle, le diremos que le interface Centronics de Indescomp es perfectamente compatible con el Interface 1 de Sinclair, ya que este último no hace nada durante el proceso de inicialización.

Por otro lado, el interface de Indescomp está bastante bien pensado, de forma que los problemas de inicialización, que se pudieran producir con otros periféricos, son fácilmente solventables. Este interface vuelca su software en RAM al conectarle la alimentación, pero si, luego, se hace un RESET (PRINT USR 0), el software desaparece y el interface no molesta para nada.

Nosotros utilizamos conjuntamente el Interface 1 de Sinclair, el Centronics de Indescomp y el Beta-Disk que también utiliza la inicialización. Entre estos dos últimos surge un conflicto que se resuelve pulsando el RESET y cargando el software del Indescomp desde disco.

«A tiros»

¿Cómo se puede hacer que un objeto dibujado en la pantalla pueda disparar?

¿En Basic se puede hacer que un objeto siga a otro, dibujados amdos sobre la pantalla y manejando sólo uno de ellos?

Alberto SANTANA - Guadalajara

☐ Para mover cualquier objeto por la pantalla, es necesario irlo borrando e imprimiendo mientras se van alterando sus coordenadas; por tanto, todo el problema se reduce a ver el modo de alterar las coordenadas para obtener un determinado efecto.

El disparo se simula muy bien PLOTeando y borrando (PLOT INVERSE 1) un pixel mientras se incrementa o decrementa una de sus coordenadas mediante un bucle FOR...NEXT.

Para hacer que un objeto siga a otro, es necesario alterar las coordenadas del perseguidor en función de las del perseguido. Veamos un ejemplo; supongamos que «li» y «co» son, respectivamente, las coordenadas «línea» y «columna» del objeto perseguido; y «l» y «c» las del perseguidor; básicamente, podríamos hacer algo así:

```
IF li>l THEN LET l=l+1
IF li<l THEN LET l=l-1
IF co>c THEN LET c=c+1
IF co<c THEN LET c=c-1
```

«QUICKSHOT V»

Os agradecería que me dijerais si es o no compatible el joystick QUICKSHOT V con el Spectrum pues, basándome en la información que dabais en el número 54, sección «Micropanorama», lo compré y resultó no ser compatible. En el comercio donde lo compré llamaron al distribuidor de barcelona y les contestaron que no era compatible y cuál fue mi sorpresa cuando en el número 58, en la sección «Expansión» volvéis a decir que es compatible.

Si, efectivamente, no es compatible, quisiera saber que cambios hay que hacer en la circuitería interios.

J. M. GONZALEZ - Pamplona

☐ Nosotros hemos probado el joystick QUICKSHOT V y ha funcionado perfectamente con el Spectrum; por ello, hemos afirmado y seguimos afirmando que es compatible. La única particularidad de este joystick es que tiene tres botones de disparo, dos rojos y uno amarillo; los rojos funcionan normalmente, pero el amarillo no puede ser leído por el Spectrum; este pequeño inconveniente tiene fácil solución si se abre el joystick y se coloca este botón en paralelo con los otros dos.

Teclados

Tengo un Spectrum 48k y desearía modificar su teclado, por lo cual, mi duda es si sería mejor comprarme el teclado Indescomp, o por el contrario, convertir mi Spectrum en un Plus.

José C. APARICIO - Córdoba

□ La elección de un teclado (como la de un ordenador), es una decisión que hay que tomar teniendo en cuenta lo que se quiere hacer con él. Existen más teclados además del Indescomp; nuestro consejo es que los pruebe todos o, al menos, los más posibles antes de decidirse.

De todas formas, el Indescomp es un teclado excelente con teclas de muy buena calidad, aunque no tiene teclas desdobladas (las numéricas y los cursores actúan en paralelo sobre las correspondientes de las dos semifilas superiores) le sobra la tecla derecha de «SPACE» que debería haber sido «CAPS SHIFT», y se echan a faltar dos teclas de «punto» y «coma» en el teclado numérico que, además, hubieran sido muy fáciles de desdoblar, simplemente, con cuatro diodos. En resumen, se trata de un teclado fabricado con buenos materiales pero con un diseño electrónico no todo lo cuidado que sería de desear. Como ventaja adicional, incluye amplificador de sonido, «RESET» y salida de monitor.

Todo lo contrario hay que decir del Plus, tiene un diseño muy cuidado y gran número de teclas desdobladas, pero se trata de un teclado de membrana y, además, mala; por lo tanto, lo que las averías están a la orden del día. En este caso hay que decir que podrían haberse gastado un poco más en mejorar los materiales de construcción, aunque esto hubiera supuesto un incremento del precio.

Interface programable y «FILL»

1. Con un joystick Spectravideo y un interface programable, ¿puedo jugar a todos los juegos?

2. En Basic, ¿cómo puedo llenar un círculo de color?

Gem ROMERO - Barcelona

□ Efectivamente, con un interface de joystick programable podrá jugar a todos los juegos; no tiene más que programar el interface para que actúe sobre las teclas que controlen el juego en particular.

Respecto a su segunda pregunta, algunos ordenadores incluyen en su Basic el comando «FILL» que permite llenar de color cualquier figura geométrica cerrada. No es el caso del Spectrum, por lo que, en principio, no es posible. Claro que siempre se podría diseñar una rutina en Basic que, haciendo uso de «PLOT» y «POINT», rellenará una figura de tinta, pero sería exasperadamente lenta. Lo mejor es recurrir al código máquina, en la MICROFICHA R-8 tiene una rutina de «FILL» que le permitirá hacer lo que quiere.

Rótulos gigantes

Tengo interés en obtener una forma de creación de caracteres alfabéticos que me permita unos rótulos con letras más grandes que los que tiene el Spectrum. Intenté utilizar el programa que publicaron en un número de su revista de hace unos meses en el que debía cargarse el código máquina de la cinta «HORIZONTES» y no lo he conseguido. Por ello, les agradecería me informaran si existe algún programa de creación de caracteres, que fuera fácilmente utilizable dentro de cualquier otro programa, para hacer las letras más grandes.

Pedro FERRE - Tarragona

□ No nos explica porqué no puede utilizar el programa de la cinta «HORIZONTESZ»; y no lo entendemos; porque, si sigue al pie de la letra las instrucciones que damos, no debe tener ningún problema.

De todas formas, en la MICROFICHA R-18 (MICROHOBBY número 61), hemos publicado otra rutina en código máquina que lo hace y que, además, podrá incluir dentro de cualquier programa.

¡No se preocupe!

Estoy haciendo un juego en el cual una de las partes esenciales es una pantalla que he hecho con el programa «SUPER-GRAFICOS». Me gustaría mandároslo, pero no sé cómo puedo enviaros la pantalla para que los demás lectores la puedan copiar. La pantalla está grabada con CODE 32768,6912.

José I. ALMUZARA - Bilbao

□ Efectivamente, sería un problema reproducir un listado de 6912 bytes, ¡una pantalla en DATAs! Precisamente, pensando en este tipo de problemas, se creó MICROHOBBY CASSETTE. No se preocupe, mándenos su programa y, si no puede ir en MICROHOBBY SEMANAL debido al problema de reproducir la pantalla, irá en MICROHOBBY CASSETTE.

DE OCASION

● NECESITO vender urgentemente un teclado Indescomp Multifunción I (cursores, amplificadores, interruptor on-off). Precio 9.000 ptas., con su caja original. Llamar al Tel. (957)234266. Preguntar por Eloy. Tardes.

● VENDO Spectrum 48k, en buen estado por sólo 23.000 ptas. Incluyo cables, transformador. Interesados escribir a Antonio Suárez Prieto. Sen Bernardino, 15. Madrid. Tel. 2481461.

● VENDO cassette especial para ordenador marca Computone, en perfecto estado, con garantía y valorado en 7.000 ptas. por sólo 4.000 ptas. (negociables), o cambio por material informático a convenir. Llamar al Tel. (985)326636 de Gijón y preguntar por María.

● VENDO los 50 primeros números de MICROHOBBY por 4.500 ptas. Regalo tapas para guardar los 20 primeros números y su índice, cassette Hobby Suerte, discoflex con programas, microfichas con rutinas, cursillo Código Máquina y posters de Gremlins. Interesados pueden llamar al Tel. (91)8502386 (tardes de 7,30 a 9). Andrés.

● VENDO consola de videojuegos Interton Electronic VC-4000, incorporado con dos mandos más 5 cartuchos de regalo, está en buen estado. Precio 10.000 ptas. También vendo Joystick Quick Shot II, casi sin usar. Precio 1.900 ptas. Todo por 11.400 ptas. También se vende por separado. Interesados llamar al Tel. (91)2080040. Preguntar por José Miguel.

● DESEO contactar con usuarios de ZX Spectrum 16/48k, para intercambio de trucos, ideas. Contactar con Miguel Madinabeitia. Escuelas, 6, 2.º Victoria (Alava).

● SE VENDE ordenador ZX Spectrum 16k en buen estado, con todos los cables. El precio es de 19.000 ptas. Interesados llamar al Tel. (985)573877. Preguntar por Beza.

● VENDO ordenador ZX Spectrum 48k, con fuente de alimentación y todos los cables, conexiones, cinta Horizontes, libros en castellano. Cassette especial para ordenador, amplificadores sonido, joystick, interface y maletín para colocar el Spectrum más sus componente. Precio 48.500 ptas. Regalo número de revistas del tema. Interesados dirigirse a Luis Nogues. Padre Gil, 3. Reus (Tarragona). Tel. (977)320979.

● VENDO ordenador Laser 310 con 16k, con alimentador, cables, instrucciones y un libro de Basic completo, todo en perfecto estado. Urge vender, por solo 14.000 ptas. Ponerse en contacto con Antonio Ojeda. Ancha, 4 ó bien llamar al Tel. (954)742152 a partir de las 3 de la tarde.

● Si alguien tiene el número 6 de Microhobby Semanal y no piensa encuadernar la parte central (coleccionable), que me lo envíe a la siguiente dirección: David Delgado. Ardale, 3, 3.º. Blanes (Gerona). Pago 300 ptas.

● VENDO Spectrum 48k en buen estado con toma de vídeo y reset instalados. Cassette reproductor especial para ordenador con todos sus conectores. TV. B/N de 14". También vendo Mini-cadena stereo con marcador. Altavoces separados. Regalo auriculares de buena calidad con selector de volumen y alargo de cable. Todo por 15.000 ptas. El ordenador completo lo vendo por 40.000 ptas. Interesados llamar al Tel. 3111047 a partir de las 9 de la noche, o bien escribir a la siguiente dirección. Segre, 112, Esc. B, Aº 2.ª. Barcelona.

● ¿NECESITAS una pantalla para tu juego? ¿Nuevos tipos de letras, gráficos? Llama al Tel. (985)848213 ó bien escribe a Juan Prada Blanco. La Plaza, 13, 1.º. Cangas de Onís (Asturias).

● CLUB Sedemur especialistas en hardware. Ampliaciones a 80k bytes, Interface Kampston reset y todo tipo de montajes. Hacemos pequeñas reparaciones. Interesados llamar al Tel. 245223 de Murcia.

● COMPRO/VENDO/CAMBIO todo tipo de hardware, así como formar un club para el intercambio de ideas, trucos, revistas de Madrid: Leganés, Carabanchel, Villaverde, Móstoles, Fuenlabrada, Alcorcón, Getafe, etc. Dirigirse a la siguiente dirección: Carlos Nevado Pérez. Avda. Menéndez Pidal, 2, 6.º C. Zarzaquemada (Leganés - Madrid).

● VENDO ZX Spectrum 48k, con todos los cables, cuatro meses de garantía, manuales en castellano, joystick, interface, todo por 33.000 ptas. Interesados pueden escribir a la siguiente dirección. Jesús Arrebola Reina. Enrique González, 4, 4.º A. Algarrobo-Costa (Málaga).

● VENDO videopac Philips G-7400 con dos cartuchos y talones descuento, o cambio Spectrum 48k o Spectrum Plus. Interesados llamar al Tel. 7984534. Madrid. Javier Gómez.

● VENDO Interface programable Investrónica, en perfecto estado, garantía en blanco, y también joystick, compatible con todos los juegos del mercado. Precio a convenir. Interesados llamar al Tel. 4472985 de Madrid. Preguntar por Fernando.

● VENDO ZX Spectrum 16k, cables, transformador, 3 libros de juegos, 1 libro para aprender a programar tus programas y uno de Basic. Interesados llamar el Tel. (96)2417238. Preguntar por Manuel.

● VENDO Spectrum 48k, en perfecto estado con todos los cables y fuente de alimentación más manual de Basic y sus correspondientes cintas. Todo por 29.992 ptas. Interesados llamar al Tel. (91)7395395. Preguntar por Marcelino García.

● VENDO amplificador para Spectrum con o sin altavoz, por sólo 2.000 ptas. También vendo Interface tipo Kempston con amplificador, altavoz, todo incorporado en el interface. Interesados llamar al Tel. (964)227826 de Castellón. Preguntar por Francis.

● CAMBIO colección de revistas MICROHOBBY del número 1 al 40. Contiene cintas de concurso Hobby Suerte y Discoflex, por un joystick con su correspondiente Interface. Escribir a Santiago Sánchez Hidalgo. Hernán Cortés, 2. Tarrasa (Barcelona). Tel. (93)7808146.

● VENDO ZX Spectrum 48k completo y español, en perfecto estado, incluyo Joystick Canon alta sensibilidad, además de una video-juegoteca. Todo por 40.000 ptas. Urge la venta por cambio. Interesados llamar al Tel. (945)286449. Federico. Vitoria.

● VENDO ZX Spectrum 48k, nuevo, completo y con garantía hasta mar-86. Interface Kempston también con garantía y 30 revistas del ramo. Todo por 27.000 ptas. Interesados llamar al Tel. 4659259. Madrid.

● VENDO teclado Spectrum, incluye caja externa, teclado de goma y conexiones, todo nuevo, por sólo 3.500 ptas. Escribir a Ernesto Hernández. Vía Hispanidad, 61, Bl. 7, 1.º A. Zaragoza. Tel. (976)347799.

● DESEARIA contactar con usuarios del Spectrum para el intercambio de ideas, trucos. También me interesaría conocer a gente a la que le interese el Logo en castellano del ZX. Escribir a José Ramón Resa. Santucho, 22, 3.º D. Bilbao (Vizcaya). Tel. (94)4338029. Preguntar por Joserra.

● VENDO Spectrum Plus, completo más interface para dos joysticks con salida a monitor, más joystick Gran Capitán, libros y revistas. Todo por 39.000 ptas. Bien cuidado. Eduardo. Tel. 4604801. Madrid.

● VENDO ZX Spectrum 48k, 1 año de uso, con cables y manual en castellano y cinta Horizontes. Se incluye teclado profesional, joystick con interface, lápiz óptico, todo ello por sólo 27.000 ptas. (negociables). Escribir a Jesús González Callejas. Alpes, 211, 6.º 4.ª. Hospitalet de Llobregat (Barcelona). Tel. 3384171.

● VENDO Spectrum 48k en perfecto estado con cables y alimentador con conexión para amplificador. Regalo Interface para Joystick. Precio 35.000 ptas. Juan Rafael Cortés Salas. Tel. (952)253833. Camino del Colemnar, 37, 7.º A. Málaga.

● CAMBIO instrucciones de Psytrom, Shadowfire, Blue Max. Dirigirse a Félix Odriozola. Palacio de Narros s/n. Zarauz (Guipúzcoa).

● VENDO ZX Spectrum 48K, con fuente de alimentación, cables, cassette de demostración y garantía. Sólo por 28.000 ptas. Interesados escribir a Fco. Javier Muñoz Romero. Prolongación de Colón, 3. Santa Fe (Granada).

● INTERESADOS en formar parte de un club de usuarios de ZX, para intercambiar ideas, listados, etc. Escribir a José Angel Soriano Rodríguez. San Emigdio, 75. Almoradí (Alicante), o bien llamar al Tel. (965)700508.

● VENDO Spectrum 16K, por sólo 19.900 ptas. Interesados llamar al Tel. (94)4580962 o bien dirigirse a Oscar Casares. Erdiko, 9, 4.º dcha. Alonsotegui (Vizcaya).

● VENDO Spectrum 48k, con todos sus accesorios, 2 libros y bastantes revistas. Precio 30.000 ptas. P. Miguel. Tel. 3095874 de Barcelona.

POWER
SOFTWARE, S.A.

NAPOLES, 98, 1.º 3.ª - Tels. 232 24 61
08013 BARCELONA (SPAIN)

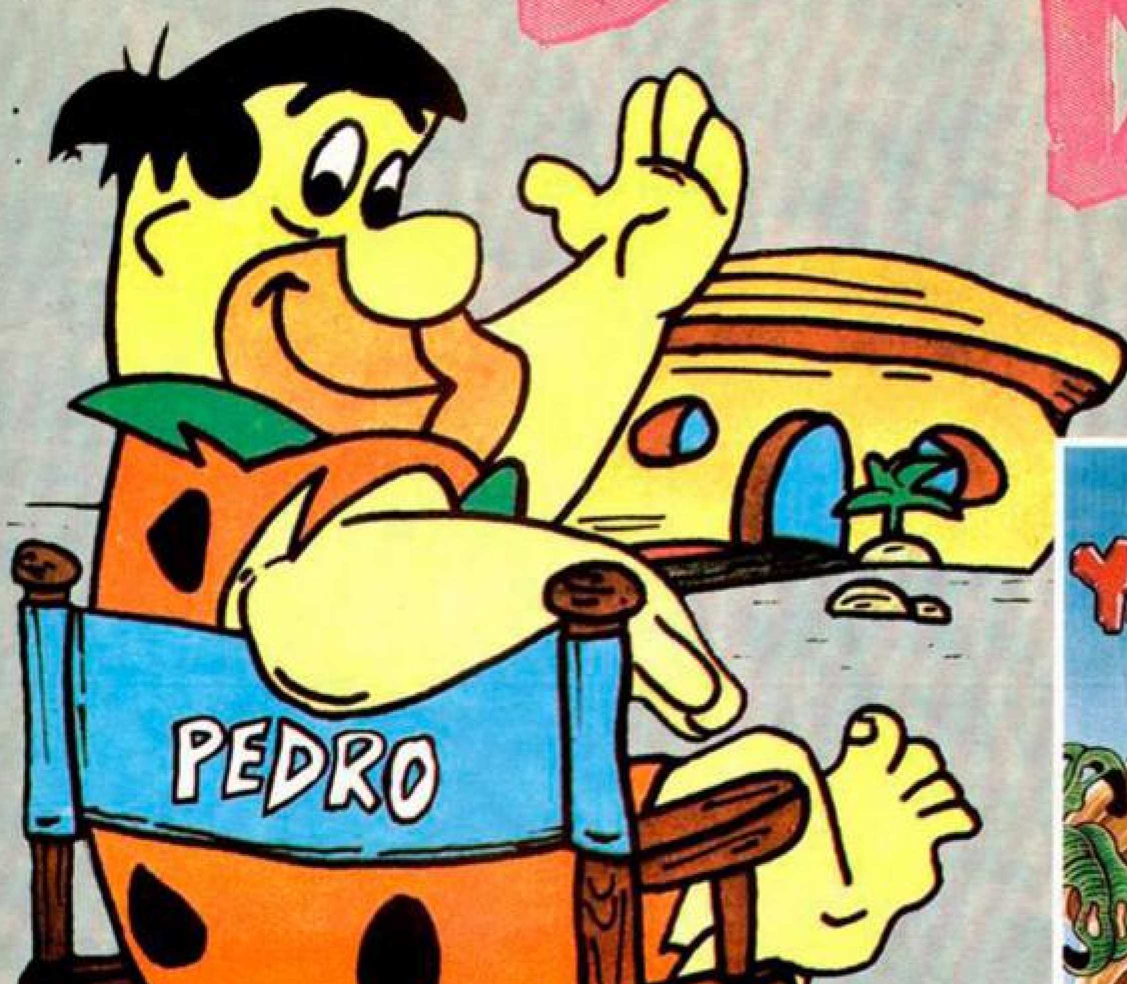
¡ATENCION! SOLO POWER SOFTWARE, S.A. POSEE LOS DERECHOS DE FABRICACION Y COMERCIALIZACION EN EXCLUSIVA PARA TODA ESPAÑA DEL PROGRAMA...

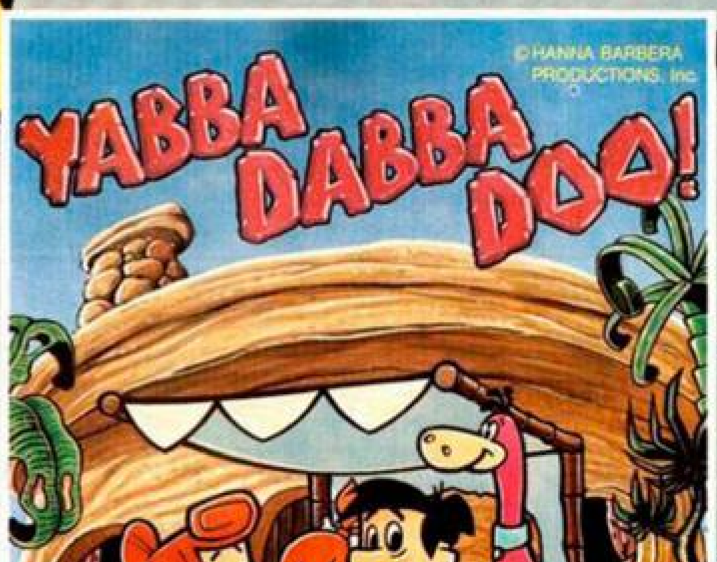
CONTIENE UN MARAVILLOSO POSTER A TODO COLOR DE LOS PICAPIEDRA Y LAS INSTRUCCIONES DEL JUEGO EN CASTELLANO.

YABBA DABBA DOO!

1.800 Pts.

COMMODORE CASSETTE
COMMODORE DISKETTE
AMSTRAD CASSETTE
AMSTRAD DISKETTE
SPECTRUM





POWER SOFTWARE, S.A. PERSEGUIRA POR TODOS LOS MEDIOS LEGALES Y/O COMERCIALES A SU ALCANCE, CUALQUIER TIPO DE PIRATERIA, COMERCIALIZACION, IMPORTACIONES O DUPLICADOS ILEGALES QUE SOBRE SUS PROGRAMAS SE PRACTIQUE.

Deseo recibir los juegos que a continuación especifico, comprometiéndome al pago del importe de los mismos.

Nombre ______________________

Dirección ______________________

Teléfono ______________________

Firma:

¡YABBA DABBA DOO!

SISTEMA ______________ CANTIDAD ______________

☐ Contrarreembolso. ☐ Adjunto Talón. ☐ Giro Postal.

Deseo recibir información de sus programas en: MSX ☐ AMSTRAD ☐